Comparative Analysis of Large Language Models for Bangla Abstractive Text Summarization

By
Md. Farhan Afsar
213-15-4292

Naimur Rahman Durjoy
213-15-4304

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Mr. Shahadat Hossain
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Mr. Raja Tariqul Hasan Tusher
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

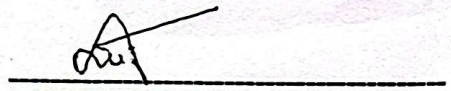
September 16, 2025

APPROVAL

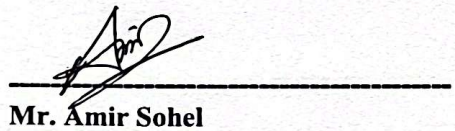
This Project titled "**Comparative Analysis of Large language Models for Bangla Abstractive Text Summarization**," submitted by **Md. Farhan Afsar** and **Naimur Rahman Durjoy** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **16 September, 2025**.

BOARD OF EXAMINERS

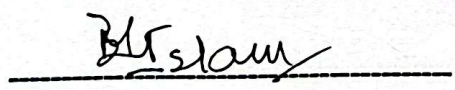
Dr. Arif Mahmud — **Chairman**
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Dr. Md. Ali Hossain — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Mr. Amir Sohel — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Dr. Md. Manowarul Islam — **External Examiner**
Professor
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Mr. Shahadat Hossain, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Mr. Shahadat Hossain
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Mr. Raja Tariqul Hasan Tusher
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Farhan

Md. Farhan Afsar
Student ID: 213-15-4292
Department of Computer Science and Engineering
Daffodil International University

Durjoy

Naimur Rahman Durjoy
Student ID: 213-15-4304
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project (FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Mr. Shahadat Hossain, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Natural Language Processing (NLP) to** carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Abstractive text summarization is a critical challenge in natural language processing (NLP), especially for low-resource languages like Bangla, where data scarcity and weak multilingual adaptation limit progress. This thesis presents a comparative study of three approaches: fine-tuned BanglaT5, fine-tuned mT5, and prompt-engineered GPT. The Bengali Abstractive News Summarization (BANS) dataset was employed, with preprocessing steps such as normalization, tokenization, padding, and truncation to ensure consistency. BanglaT5 and mT5 were fine-tuned using AdamW with cross-entropy loss, while GPT was evaluated through zero-shot prompts. Performance was measured with BERTScore and human evaluation by three annotators, who rated outputs on Relevance, Coherence, and Conciseness (1–10 scale). Automatic results show that BanglaT5 achieved the highest BERTScore (F1 0.817% in Bangla embeddings; 0.957% in English embeddings), outperforming mT5 (F1 0.551% in Bangla; 0.765% in English). Human evaluation revealed that GPT consistently scored higher in Relevance 85% and Coherence 84%, while BanglaT5 was rated better for Conciseness 88%, reflecting its ability to produce shorter yet meaningful summaries. These findings highlight the trade-offs between language-specific and general-purpose LLMs: BanglaT5 excels in conciseness and precision, GPT in fluency and relevance, and mT5 underperforms across dimensions. The study concludes that a hybrid approach, combining the precision of BanglaT5 with the fluency of GPT, can significantly advance Bangla summarization and contribute to more inclusive NLP tools for low-resource languages.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the research problem, highlights the motivation behind the work, and sets out the objectives and contributions of the study. It provides a clear context for why Bangla abstractive summarization is important and outlines the structure of the thesis.

1.1 Introduction

Natural Language Processing (NLP) has seen transformative advancements in recent years, largely driven by large language models (LLMs) and the rise of transformer-based architectures. These models have enabled remarkable progress in tasks ranging from text generation to question answering, but their benefits have not been uniformly distributed. High-resource languages such as English, Chinese, and Spanish are supported by extensive datasets and computational resources, while low-resource languages like Bangla continue to face significant challenges. Despite being spoken by over 230 million people worldwide, Bangla content on the web and in structured datasets remains relatively limited, and the work on Bangla NLP is still developing [9, 14].

This task is very challenging, especially for low-resource languages, such as abstractive text summarization, which aims to generate a short and coherent text that expresses the essence of the original text. The complexities of Bangla grammar and syntax as well as a lack of high-quality large-scale datasets are factors that impede the performance of both traditional and modern multilingual models. Previous studies, e.g., Mukherjee (2022) have demonstrated that multilingual models such as mT5 may face challenges for Bangla summarization, and sometimes perform worse than simpler architectures such as the LSTM-based models due to resource constraints and lack of language fine-tuning [5]. Recent systems have investigated the use of general-purpose LLMs, but have showed only modest results, indicating the usefulness of more specialized solutions [11].

This work aims to alleviate such limitations by introducing a comparative study of large language models for Bangla abstractive text summarization. More specifically, we fine-tune two models, the multilingual mT5 and the Bangla-specific Bangla-T5, on the BANS (best answer selection) dataset, which is one of the most popular benchmarks for the task. Our method is inspired by the advances that a language-specific, fine-tuned model can achieve over the multilingual model, reported by Mondal et. (2025) which demonstrated Bengali-T5 provided a substantial performance improvement over mT5 on a similar summarization task [13].

A significant contribution of this work is the human-directed analysis of the output of fine-turned models. We evaluate the summaries generated by our fine-tuned models against a strong, general purpose LLM, ChatGPT. Although commercial models such as GPT-4 have done well on recent benchmarks [12], to the best of our knowledge, this is the only qualitative comparison on the basis of human ratings. By having 3 expert human judges score the summaries for relevance, coherence, and conciseness, we hope to show that a specialized, highly-tuned model can produce

briefer and more targeted summaries than a language model that is powerful but general purpose. The results of this study not only serve to improve the Bangla NLP, but also highlight a favorable approach for enhancing current state of the art models to any other low-resourced language.

1.2 Motivation

This research is driven by the challenges in developing effective Natural Language Processing (NLP) systems for Bangla, a language spoken by over 230 million people worldwide. Despite its large speaker base, Bangla remains a low-resource language in NLP, with limited datasets and computational resources compared to high-resource languages like English and Chinese. As a result, advanced systems like abstractive text summarization are underdeveloped for Bangla.

In case of Bangla, abstractive summarization is a difficult task because of its complex grammar and absence of language specific model. Though models such as mT5 have shown great promise, it is still the case that due to a lack of fine-tuning, they are unable to capture the linguistic complexity of Bangla. The Banga-T5 model which is fine-tuned for Bangla is theoretically more focused, however comparison with state-of-the art models such as ChatGPT can give some intuitive reasons for why the results of Bangla-T5 are not that promising.

This study is focused on the comparison of mT5, Bangla-T5, and ChatGPT models for Bangla abstractive text summarization and fine-tuned the models over the BANS dataset. By conducting experiments on existing summarization systems, we provide oracle as well as comparative performance with both automated (and human evaluation) to inspire development of Bangla summarization systems and to shed more light on the true potential of large language models in resource-constrained languages, such as ChatGPT.

The findings of this work could further aid in the development of NLP for Bangla and contribute to the larger project of enhancing summarization systems for under-resourced languages. Additionally, this will contribute towards the making of multilingual AI systems, enabling millions of speakers around the world to have access to more accurate and efficient NLP tools.

1.3 Objectives

The main objective of this research is to conduct a comparative analysis of two large language models mT5 and Bangla-T5 for Bangla abstractive text summarization, using the BANS dataset. The specific objectives are:

- To fine-tune the mT5 and Bangla-T5 models on the BANS dataset to improve their performance in generating accurate and concise summaries of Bangla text.

- To compare the performance of the fine-tuned mT5 and Bangla-T5 models in terms of summarization quality, using automated evaluation metrics (BERTScore).

- To evaluate the effectiveness of the fine-tuned models by comparing their summaries with those generated by ChatGPT, focusing on factors like

relevance, coherence, and conciseness.

- To assess the quality of summaries produced by mT5, Bangla-T5, and ChatGPT through human evaluation, involving three professionals who will rate the summaries on relevance, coherence, and conciseness.

- To determine whether fine-tuning Bangla-T5 results in more concise and relevant summaries than ChatGPT and mT5, thus contributing to the development of better Bangla summarization systems.

- To explore the broader applications of this research in enhancing summarization systems for other low-resource languages, offering a framework for the development of language-specific NLP tools.

1.4 Methodology

The main goal of this work is to compare and contrast different large language models for abstractive text summarization on a low resource language - Bangla. Our aim in this research is influenced by the rising demand for powerful NLP resources for the Bangla language and the capability of domain finetuned models to be superior to general purpose models.

We use the BANS dataset from Kaggle [22], which contains 19,096 data points and is an intensive corpus suitable for training and evaluation. In order to tackle this problem, we used transfer learning approach by fine-tuning two transformer-based models: mT5 and Bangla-T5. These models were fine-tuned on the BANS dataset to customize them for Bangla abstractive summarization. Upon fine-tuning, a preliminary qualitative analysis showed that the Bangla-T5 model was performing better than the mT5 model while generating succinct and contextually informative summaries.

The human evaluation study was conducted to additionally evaluate our results, and to compare our fine-tuned model with a state-of-the-art general-purpose model. We randomly sampled 50 articles from the dataset, and generated both offline summaries with fine-tuned Bangla-T5 and an off-the-shelf state-of-art commercial large language model ChatGPT on each of them. These summaries were then assessed by a three-member panel of professionals with a relevant academic background. They were prompted to rank each summary (on a scale of 1 to 10) according to three dimensions: Relevance, Coherence, and Conciseness. The results of this analysis suggested that our fine-tuned Bangla-T5 model has a better balance between conciseness and readability on the summaries when compared to ChatGPT.

Contribution and Benchmark Here we contribute to the field of Bangla NLP by showing the effectiveness of fine-tuning the language-specific models and provide a valuable benchmark through a very well-organized human evaluation process.

1.5 Project Outcome

The outcome of this research is expected to significantly advance the field of Bangla Natural Language Processing (NLP) by addressing key challenges in Bangla abstractive text summarization. The primary outcomes of this project include:

Improved Bangla Summarization Models: By conducting a comparative analysis of large language models such as mT5 and Bangla-T5 for Bangla abstractive text summarization, this project aims to enhance the quality and relevance of summaries produced by these models. The research is expected to demonstrate that fine-tuning a Bangla-specific model, such as Bangla-T5, will yield better performance compared to multilingual models like mT5.

Building Benchmark for Bangla NLP: This research will establish a shared evaluation system for Bangla abstractive text summarization, in particular by further fine-tuning models on the BANS or other dataset. This framework will act as the initial stage for further work in the area, to help compare other models and methods that can be used to improve summarization for less resourced languages.

Human summary perception: This work contributes significantly to human evaluation of summaries. By enlisting expert raters from this specific domain to judge the adequacy of the summaries in terms of relevancy, cohesion and conciseness this research attempts to provide observational qualitative findings that may be neglected by the quantitative indices. This will provide an insight into the degree to which these models actually address the requirements of real-life summarization.

Impact on Low Resource Language NLP: This work is expected to support the overall goal of the NLP community to develop NLP technologies for low-resource languages including Bangla. Using this implementation, we continue to fine-tune models to develop targeted solutions for Bangla and by extension, we hope to overcome the limitations of NLP resources available for non-English languages and promote the replication of the same for other low-resource languages as well.

Applications to other fields: The findings of this project will have wide-ranging implications in a number of practical applications such as news aggregation, educational tools and automated summaries to content. Automatic summarization of Bangla text can be used to make content accessible as well as to improve efficiency in areas such as media, education and government services.

Contributions to Multilingual AI Systems In addition to these contributions, the results of this work can potentially contribute to the construction of future multilingual AI systems by helping to understand the benefits brought by language specific instead of multilingual models. This has the potential for promoting more advanced AI systems that can better serve and accommodate different linguistic needs, including for low-resource languages such as Bangla, in solving these tasks.

In summary, the results of this proposed work will result in advancing Bangla NLP as well as it will lead to developing scalable approach to bettering summarization system for low-resourced languages, allowing digital inclusivity, innovation.

1.6 Organization of the Report

This report is divided into the following sections:

Chapter 1 Introduction This chapter introduces the research study and focuses on the background to the study, the rationale, and presents the main aims and objectives of the study. It also mentions the anticipated outcomes of the project, and how it can enrich the domain of Bangla NLP.

Chapter 2 presents a comprehensive literature survey about NLP, text summarization methods, and works previously done on models e.g., mT5, Bangla-T5. The chapter addresses the existing problems in Bangla NLP, particularly for summarization and describes the prior work and approaches for summarization. A gap analysis is also performed to identify the shortcomings of existing methods and to justify the need for this research.

Chapter 3 describes the proposed methodology of the research undertaken to realize the developed project that comprises with models selection, data gathering procedure, and models fine tuning method. It reports on the different evaluation metrics, ranging from automatic (ROUGE, BLEU) to (human-aware) human-based (relevance, coherence, conciseness) measures to evaluate the performance of the models.

Chapter 4 details the experimental setting environment and the procedures used to develop the Bangla-T5 and mT5 model for text abstractive summarization. We show the results of the model evaluation, qualitative and quantitative, too. The chapter also juxtaposes the performance of the two models and talks about the limitation and strengths of the models for Bangla summarization in practical scenario.

The engineering practices are described in Chapter 5, covering best practices in software development, setup of the hardware, and communication between nodes. It also discusses the design issues that have to be faced, the ethical issues concerning the research and the social impact of the proposed solution.

Chapter 6 gives a conclusion of the findings in the study, limitations of the study, and directions for future work. The chapter further discusses the potential impact of the study on advancing NLP for low-resource languages and improving the quality of text summarization systems.

Every chapter is structured so as to ensure gradual advance from problem definition to solution design, so that the specific goals of the research are obtained and the project contributions fully presented.

Chapter 2

Background

This chapter reviews prior work in abstractive summarization, focusing on transformer-based approaches and Bangla-specific advancements. It identifies the strengths and limitations of existing models and highlights the research gaps that motivate this study.

2.1 Introduction

The study titled, "Comparative Analysis of Large Language Models for Bangla Abstractive Text Summarization", explores the ability of state-of-the-art AI models in producing connected text summaries in Bangla language. Great advancements have been made in artificial intelligence (AI) so that machines become more proficient at understanding and generating natural language, thereby upending the way people engage with digital systems. Over this changing terrain, abstractive text summarization has stood out as an important task for extracting information out of the abundant avalanches of text, but in a way that mimics human cognition and expression.

Bangla is one of the most spoken languages, still considered as a low-resource language in the domain of NLP. This digital divide restricts millions of Bangla-speaking users from intelligent technology. Previous works of Bangla summarization using the rule-based or extractive approaches have faced difficulties to produce fluent and context-sensitive abstractive outputs. But, with transformer models in place (like mT5, BanglaT5.) there lies a good hope to pursue for abstractive summaries which are conceptually rich and structurally fluent.

This paper compares mT5 vs BanglaT5 in the task of Bangla news summarization. Both models are transfer-learned on large multilingual or Bangla-centered datasets and have demonstrated success on other low-resource language tasks. For experiments, we use the Bengali Abstractive News Summarization (BANS) dataset, containing a sizable collection of article-summary pairs. ChatGPT is also added as a benchmark model to demonstrate the performance difference between a general LLM and models fine-tuned for Bangla summarization. Early ChatGPT generations are Ok for small text, but start to lose coherence for bigger paragraphs. However, the mT5 and BanglaT5 models perform, overall, better and more stably across different inputs.

The results of this study intend not only to compare the performance of models but also to the linguistic fairness of the NLP, specifically, in Bangla. By finding good summarization models for Bangla, our work enables fairer usage of AI tools for all, and also promotes linguistic diversity.

2.2 Literature Review

The task of abstractive text summarization has seen considerable progress over the last decade, from conventional statistical and heuristic strategies, to the state-of-the-art deep learning and transformer-based methods. These advancements have been particularly far-reaching for low resource languages such as Bangla, where availability of small data sets and computational resources for natural language processing create unique challenges. Recent work has sought to exploit large language models (LLMs) and transformer-based architectures for improving summary quality, as well as focusing on language-specific personalization, multilingual systems and evaluation on more varied corpora.

Sunitha et al. 2016), are among the first studies on abstractive summarization for Indian languages. The paper surveyed both extractive and abstractive approaches and concentrated on linguistic Issues such as morphology, syntax, and limited corpora. The authors highlighted the fitness of semantic graphs, template-based approaches and neural sequence-to-sequence models for Indian languages. The study, however, did not provide a new dataset but responding to the scarceness of annotated corpora and adaptation of models that were primarily made for English. Limitations lacked large-scale explorations and performance evaluations of the practicality based, hence a more rather a conceptual survey than add empirical contribution [17].

Etemad et al. (2021) introduced an early transformer-based summarization model by finetuning T5 (a transformer model) on the XSum and Gigaword datasets. Strong ROUGE scores were reported on the Gigaword dataset (ROUGE-1: 43.02, ROUGE-L: 37.43), presenting the T5's capabilities to summarization. On XSum, however, the model failed, managing only a 30.91 ROUGE-1 score. The difference in length of the output between the dataset and the model summaries played a large role for this discrepancy. The work pointed out the restriction of limited resources and that the model could not handle longer summaries well [6]. This particular limitation served as a basis for additional work focused on the problems of model output length and dataset adaptation.

Exploiting this direction, Mukherjee (2022) explored abstractive summarization in Bengali, comparing mT5 to an LSTM-based encoder-decoder model with 19,000 article-summary pairs from Bengali news articles. The results indicated that LSTM achieved better results than mT5 (an F1-score of 75% vs 56.27%). The lower performance of mT5 was ascribed to the lack of resources of the transformer model, which had difficulties with short input lengths in the dataset. However, the work by Mukherjee emphasized the potential of transformer-based architectures for Bangla summarization in the long run as dataset size and computational infrastructure gets better [5].

Agarwal et al. (2022) Introduced IndicBART to perform Hindi abstractive summarization Context ILSUM shared task. The dataset used for training the model consisted of 7,957 examples and 569 examples on validation. from Hindi news articles. Abstract: We present IndicBART, a multilingual sequence-to sequence model that comes fully pretrained. on the test, it achieved 0.544/0.443/0.400 in ROUGE-1 F1/ROUGE-2 F1/ROUGE-4 F1 respectively set. It did, however, demonstrate the usefulness of transfer learning for low-resource Hindi. restricted by the size of dataset and compute resources. Future directions proposed include better

preprocessing and longer training epochs to make the output more fluent and factual. [18]

The very year, Tawmo et al. (2022) presented an evaluation of the T5-base model on CNNDM, MSMO, and XSumference. T5-base obtained the highest scores on MSMO dataset (ROUGE-1: 42.29, BLEU: 43.9) while performing poorly on the single-sentence summaries of XSum. Shallow preprocessing and lack of fine-tuning were listed as drawbacks, the latter contributing to worse generation performance on XSum and (significantly) other datasets that have special structure requirements. This emphasized the need for heavy finetuning and dataset-specific preprocessing to reach the strongest summarization performance [7].

As the community advanced, Rehman et al. (2023) compared performance on CNN-DailyMail, SAMSum, and BillSum datasets over three models, PEGASUS, T5-base, and BART-large. Both PEGASUS and BART-large consistently outperformed T5-base, especially on the CNN-DailyMail and SAMSum datasets. Although PEGASUS was particularly finetuned for abstractive summarization, the model has relative higher performance in ROUGE metric on each of the datasets which further demonstrates its effectiveness in both encoding news articles and dialogues. However, Rehman et al. indicated that the absence of per-dataset tuning, the limitation in GPU resources, and the limit in input length affected overall performance of the models To address these issues, we first compared the performance of a model trained on the P100 (based on; ed on a smaller dataset) with that trained on a V100 since it yields the best performance under other circumstances, as Using a P100 GPU will be more practical[3].

In continuation to the above, Satya et al. (2023) compared different T5 variants for Indonesian summarization against the INDOSUM dataset. Under all such conditions, best performance was achieved by T5-Base (ROUGE-1: 73.52), with FLAN-T5 in mid position and mT5 delivering lowest results. The errors in the summaries were also indicative: T5-Base suffered with redundancy, FLAN-T5 with truncation issues, and mT5 with factual discrepancies. This work pointed out that, even with state-of-the-art models such as T5, issues with model architecture and fine-tuning are far from solved, particularly for languages with fewer resources [4].

In Bangla specific developments, Khan et al. (2023) presented the BanglaCHQ-Summ dataset for Bangla consumer health question summarization consisting of 2,350 Q-S pairs. It was observed that finetuning of the BanglaT5 model on this dataset resulted in a ROUGE-L value of 48.35%, superior to multilingual models like mT5 and mBART. The performance was promising but the small data size and large-scale fine-tuning computation cost were two major obstacles. This paper put an emphasis on the use of task-specific datasets for enhancing the quality of domain-specific summarization [2].

Simultaneously, Ilanchezhiyan et al. (2023) developed a translation-based pipeline for summarizing in multiple languages, including Hindi, Bengali, Gujarati and English, with T5-base. The study reported only moderate ROUGE and BERT F1, and suffered from problems such as code-mixing, script diversity, or the subtle structures of various languages. This work highlighted the importance of pretraining in the native language when the task is to be performed in the native language, and showed that the use of multilingual models without language fine-tuning may not be enough for addressing complex language phenomena [8].

Lal et al. Developed abstractive summarization in a low-resource setting for Hindi Abstractive Summarization for Hindi in low resource setting. They conducted an experiment involving three encoder-decoder models: BASE (attention-based), MED (multi-level encoding), and then RETRAIN (transfer-learning from English Gigaword) A Hindi Text Short Summarization while the HTSS corpus was sampled from 5k, 20k and 100k sizes. Results showed poor performance at 5k and 20k, and outputs that are repetitive and nonsensical. RET at 100k topples BASE and Motivation-Enhanced Dialogue (MED), which obtains ROUGE-1 F1 25.5, ROUGE-2 F1 7.9, and ROUGE-L F1 23.8, respectively. A new metric, ICE-H, which was not consistent itself because Hindi was not well enough embeddings and POS taggers. The study suggests: Data scarcity and resource scarcity as key barriers [16].

Nishant et al. (2023) also evaluated the performance of mT5-small and mT5-base on Indian language summarization tasks, namely, Bengali and Odia. mT5-base was superior for Bengali summarization but inferior for Odia possibly since the data was scarce. The work emphasized the significance of language specific pretraining, as well as the issues of hallucination, inaccuracy, and noise induced by the shortage language-specific resources [9].

By 2024, the focus had largely shifted to evaluating LLMs for Bangla summarization. Kabir et al. (2024) provides the BenLLM-Eval benchmark for measuring GPT-3. 5, and LLaMA-2 on seven Bangla NLP tasks. GPT-3. 5 and Claude-2 had moderate summarization ability, (ROUGE-1 ≈ 20)) and LLaMA-2 performed worst, likely due to its English-focused pretraining. The tasks also include a specifically selected question-answering (QA) task and a chunking task in order to provide relation extraction without treebank parsing in [11] Although results are only modestly successful, challenges in attempting to generate consistent output across the different tasks are reported, stress the demand for language-specific adaptation.

Rony and Islam (2024) conducted an evaluation of several LLMs, including GPT-3.5, GPT-4, OPT, LLaMA-2, and PaLM-2, for Bangla news summarization using the BANS and BNLPC datasets. GPT-4 consistently outperformed the other models in terms of coherence, faithfulness, and relevance, performing closest to human summaries. However, the study was limited by sample size and dataset diversity, suggesting that broader evaluations would be needed to generalize these findings [12].

Nguyen et al. (2024) compared summarization evaluation among patent documents. They evaluated seven models (T5, XLNet, BART, Pegasus, BigBird, LongT5, GPT-3. 5) using eight automatic metrics, human judgments, and a new LLM-based approach. Findings revealed that common metrics such as ROUGE-2, BERTScore, and Sumac had low correlation with human total23 judgments, however GPT-4 and Llama-3 evaluations are strongly correlated (corr >0.8). Human and LLM scores ranked GPT-3. 5th best for clarity, accuracy, coverage, and XLNet, and BART close behind. The study also presented an LLM based iterative refinement formulation feedback (which in turn improves clarity and coverage but decreases slightly the accuracy. This work highlights LLMs have the potential of being inexpensive, but reliable evaluators [19].

Significant progress has also been made in recent work in 2025. Mitria et al. (2025) carried out Khasi summarization with RoBERTa as the encoder and GPT-2 as the

decoder. Even with reasonable ROUGE scores (ROUGE-1: 0.37, ROUGE-2: 0.12), the system struggled with proper nouns and hallucination. This article focused on the limitations of data size and the necessity of larger and more complete research data [1].

Fang et al. (2025) explored a collaborative multi-LLM framework involving GPT-3.5, GPT-4o, and LLaMA3-8B. This framework outperformed single-model baselines by up to threefold (ROUGE-1: 0.479), showcasing the power of combining multiple LLMs. However, scalability and high computational costs remain significant barriers to widespread use [10].

Mondal et al. (2025) developed Bengali-T5 (bT5-base), fine-tuned on 20,000 Bangla news articles. This model showed dramatic improvements over mT5, achieving a ROUGE-1 score of 55.63% compared to mT5's 2.49%. This result demonstrated the value of task-specific pretraining for Bangla and proved that larger, language-specific datasets significantly boost performance [13].

Complementing this, Mahmud et al. (2025) reviewed similarity analysis methods such as Word2Vec and Doc2Vec for Bangla, noting their potential in enhancing summarization quality. However, gaps in lexical resources and neural embeddings still pose challenges to further advancements [14].

Finally, Eftee and Abrar (2025) benchmarked open-source LLMs, including Mistral, DeepSeek, LLaMA, and Gemma, for multi-document Bangla summarization. Mistral-8x22B performed the best, with a ROUGE-1 score of 28.68 and BERTScore-F1 of 0.7454. Despite its strong performance, issues of consistency, scalability, and handling long summaries persisted, demonstrating the need for further improvements in model robustness [15].

In conclusion, the literature demonstrates a clear trajectory from early transformer models to LLM-driven approaches in Bangla and low-resource summarization. While multilingual models such as mT5 have laid the groundwork, Bangla-specific models like BanglaT5 and Bengali-T5 consistently outperform them, reflecting the importance of task- and language-specific pretraining. Collaborative frameworks and open-source LLMs have broadened horizons, but challenges remain in data scarcity, computational efficiency, handling of long-form inputs, and language-specific expressiveness. Future research must focus on developing richer Bangla datasets, optimizing fine-tuning strategies, and exploring multi-model collaborations to further improve the quality and reliability of abstractive summarization systems for low-resource languages.

Table 2.1: Summary of Literature Reviewed.

Author (s)	Year	Title	Methodology	Key Findings
Eftee and Abarar	2025	Multi-Document Summarization for Bangla News Using Open-source LLMs	Evaluated eight open-source LLMs for Bangla MDS on BUSUM-BNLP dataset using ROUGE, BLEU, and BERTScore.	Mistral-8x22B outperformed others in ROUGE-1 and BLEU-4; challenges with summary length and language consistency.
Mahmud et al.	2025	Similarity Analysis in Bangla Text: A Systematic Review	Reviewed methods like Word2Vec, LDA, and WordNet for measuring similarity in Bangla text.	Word2Vec and Word Mover's Distance were most effective; need for a comprehensive Bangla WordNet.
Mondal et al.	2025	Bengali Abstractive Summarization with mT5 vs Bengali-T5	Compared mT5 and Bengali-T5 on 20,000 Bengali news articles using ROUGE and Exact Match.	Bengali-T5 outperformed mT5 in all metrics, highlighting the importance of task-specific pretraining.
Fang et al.	2025	Multi-LLM Summarization Framework for Long Documents	Proposed multi-LLM framework using GPT-3.5, GPT-4o, LLaMA3-8B for long-documents datasets (ArXiv, GovReport).	Multi-LLM outperformed single-LLM; high computational cost for decentralized approach.
Satya et al.	2025	Comparative Analysis of T5 Variants for Indonesian Summarization	Evaluated T5-Base, FLAN-T5, mT5 on INDOSUM dataset using ROUGE metrics.	T5-Base performed best, mT5 showed low accuracy; trade-offs between speed and accuracy.
Mitria et al.	2025	Abstractive Summarization for Khasi Language	Fine-tuned RoBERTa and GPT-2 on 10,906 Khasi article-summary pairs.	Moderate ROUGE scores, issues with named entities and hallucinations.

2.3 Gap Analysis

The literature review reveals that significant progress has been made in abstractive summarization with transformer-based and large language models. However, when focusing specifically on Bangla summarization, several key research gaps persist, which restrict the development of robust and generalizable systems.

One of the biggest limitations is lack of using Bangla specific finetuning transformer models." Studies like Etemad et al. (2021), Tawmo et al. (2022), and Rehman et al. (2023) demonstrated the potential of pretrained transformer models, but focused mainly on the English or other high-resource language. This restricted their porting to Bangla, in which linguistic morphology and syntax are highly divergent.

Another gap is the underperformance of multilingual models, such as mT5 and mBART. While multilingual pretraining addresses this for several languages, studies like Mukherjee (2022), Satya et al. (2023), Khan et al. (2023), and Nishant et al. (2023) consistently and repeatedly noted that mT5 performs worse on Bangla tasks. This result underscores what is often the case with multilingual models which get only so far in matching the depth of linguistic representation that Bangla needs, which ultimately reduces the semantic and factual faithfulness.

The third gap lies in the limited exploration of GPT and LLMs for Bangla summarization. While Kabir et al. (2024) and Rony & Islam (2024) included GPT-3.5, GPT-4, Claude, and other LLMs in their evaluations, the results were modest and limited by English-centric pretraining and small evaluation samples. More recent work by Eftee and Abrar (2025) extended this evaluation to open-source LLMs such as Mistral and LLaMA but still lacked Bangla-specific adaptation and rigorous testing. This indicates that the potential of GPT and other LLMs for Bangla remains underexplored.

A further gap is the absence of human evaluation in most studies. Many works relied solely on automatic metrics such as ROUGE and BLEU, which fail to capture subjective qualities like readability, coherence, or conciseness. Only a few attempts (e.g., Rony & Islam, 2024) included small-scale human assessment, but these lacked systematic design and sufficient sample size. This creates a critical gap in evaluating how well models align with human expectations.

Table 2.2: Analyzing Research Gap

Author	Pretrained Model	Multilingual Model	GPT	Human Evaluation
Mondal et al. (2025)	✓	✓	×	×
Eftee & Abrar (2025)	✓	✓	✓	×
Rony & Islam (2024)	×	×	✓	✓
Nishant et. al. (2023)	×	✓	×	×
Our Proposed Study (2025)	✓	✓	✓	✓

2.4 Summary

This chapter has extensively analyzed the literature and the state of the art on Bangla abstractive text summarization. It highlighted drawbacks of existing multilingual models like mT5 in modeling the intricate nature of Bangla language, including morphology and syntax. The paper explained the necessity of such dedicated models such as Bangla-T5, that are fine-tuned to handle such complexities, to obtain improved summarization performance. It also investigated the importance of pre-training very large language models for low resource languages, Bangla in this case along with the transfer task and a deeper study of existing literature and models. The chapter also provided a gap analysis of the current approaches which have motivated of this work, indicating which issues are problematic, and the need of this research to fill such gaps; particularly, the improvement of the quality of summarization and the performance of the model for Bangla.

Chapter 3

Research Methodology

This chapter describes the methodology adopted for the study, including dataset preparation, model selection, training strategies, evaluation techniques, and system design. It explains the step-by-step process followed to implement the proposed approach.

3.1 Methodology

3.1.1 Overview

This study proposes a comparative analysis of large language models (LLMs) for Bangla abstractive text summarization, focusing on BanglaT5, mT5, and GPT-based models. Abstractive summarization is the process of generating concise and semantically meaningful summaries by understanding and rephrasing the core content of the input text, which is particularly challenging for low-resource languages like Bangla.

The study employs the Bengali Abstractive News Summarization (BANS) dataset, which consists of Bangla news articles with human-generated summaries. First, we fine-tuned two encoder-decoder based transformer models, BanglaT5 and mT5 on this dataset using supervised learning. Automatic evaluation the models were evaluated with common-standard automatic measures including BERTScore implemented with Hugging Face Transformers library and PyTorch.

In the second stage of the work, we considered a prompt-based summarization method with the GPT model (GPT-4o). By creating domain-specific prompts, Bangla summaries were generated without further fine-tuning. Allowing benchmarking with fine-tuned models (eg: BanglaT5, mT5) and zero-shot prompting with GPT.

For the quality of summaries results, a human judgement was done based on three fundamental dimensions Coherence, Relevance, Conciseness. With a mixed automatic human evaluation approach, this work attempts a comprehensive comparison between domain-specific, multilingual, and generative LLMs on Bangla text summarization.

3.1.2 Proposed Methodology

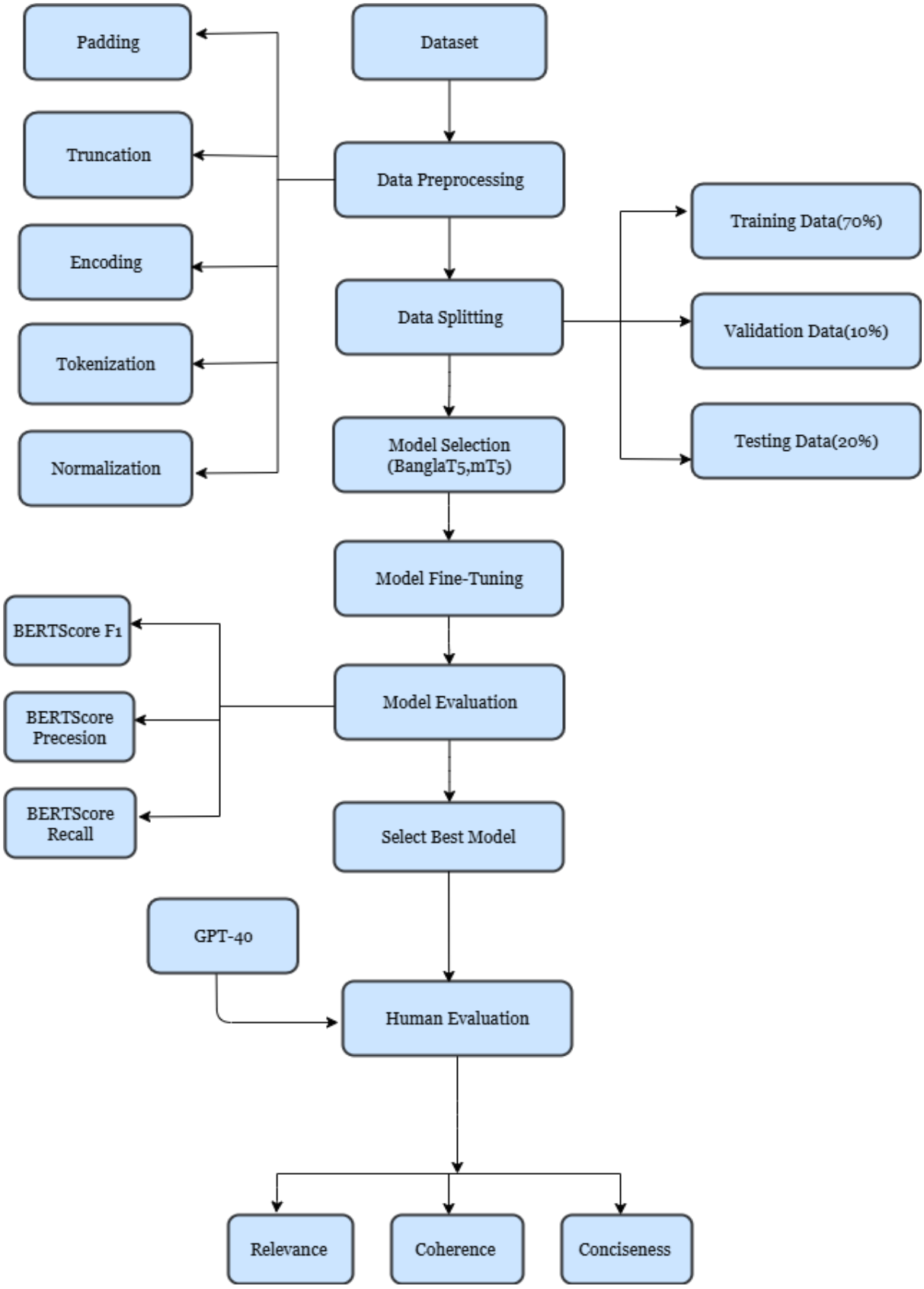


Figure 3.1: Proposed Methodology

3.2 Detailed Methodology and Design

1. Data Collection and Analysis:

In this research, the primary dataset used is the Bengali Abstractive News Summarization (BANS) dataset, which was collected from Kaggle. The dataset is publicly available and widely used for Bangla language summarization tasks. It consists of pairs of Bangla news articles and their corresponding human-written summaries.

Each entry in the dataset includes:

Table 3.1: Dataset Overview

Field	Description
article	Original Bangla news article text
summary	Human-written abstractive summary in Bangla

To provide a better understanding of the dataset's composition, key statistics of the BANS dataset are summarized in Table 3.2. These include the number of articles and summaries, as well as the range of word counts in both fields.

Table 3.2: Key Characteristics of the Dataset

Total No of Articles	19,096
Total No of Summaries	19,096
Maximum No of Words in an Article	76
Maximum No of Words in a Summary	12
Minimum No of Words in an Article	5
Minimum No of Words in a Summary	3

A representative sample from the dataset is provided below to demonstrate the nature of the content:

Table 3.3: Dataset Sample

Article	Summary
স্ট্যান্ডার্ড চার্টার্ড ব্যাংকের নতুন প্রধান নির্বাহী কর্মকর্তা হিসেবে দায়িত্ব পেয়েছেন আবরার এ আনোয়ার।	স্ট্যান্ডার্ড চার্টার্ডের নতুন সিইও আবরার
রাজধানীর মোহাম্মদপুরের একটি বস্তিতে আগুনে দগ্ধ হয়ে চার বছরের এক শিশুর মৃত্যু হয়েছে।	মোহাম্মদপুরে বস্তিতে আগুনে শিশুর মৃত্যু

জাতীয় চলচ্চিত্র পুরস্কার ও একুশে পদকজয়ী নির্মাতা চাষী নজরুল ইসলাম রাজধানীর একটি হাসপাতালে চিকিৎসাধীন অবস্থায় মারা গেছেন।	চাষী নজরুল ইসলাম আর নেই

This sample reflects the abstractive nature of the task. The summary is not an exact extract but rather a concise reformulation of the core idea in the article.

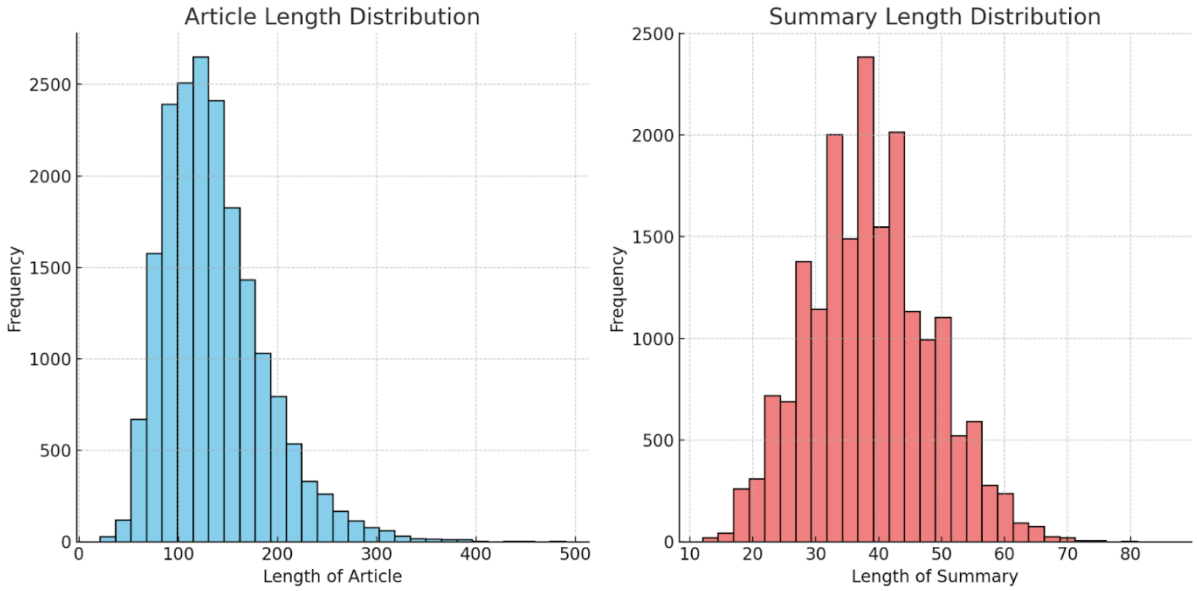


Figure 3.2: Summary Length Distribution

In order to interpret the dataset further, a statistical analysis of the length of articles and summaries has been performed. The distribution of summary lengths represented in Figure 3.2 indicates that the most of the summaries are centered around 40 characters in length, the 30-40 characters range is the densest one, and a small percentage of the summaries are between 100 and 200 characters long. Finally, article lengths are distributed more widely, but with a peak around 80-160 characters and decreasing thereafter. The distribution indicates that the summaries are uniformly extractive, in the sense of being abstract, rather than extractive. This is consistent with human-written summaries in news applications and underlines the necessity to train models with the competence to compress and reformulate long documents. From this analysis, sensible maximum lengths were chosen during the tokenization and model configuration step in order to guarantee efficient and context-sensitive summary generation.

2. Dataset Preprocessing

In this study, several pre-processing steps were conducted to transform the BANS dataset ready to be served for train in both BanglaT5 and mT5 models. These were critical in order to handle the linguistic and structural variance of Bangla text and transformed the raw data in a clean and machine-readable manner suitable for neural sequence-to-sequence modeling.

Cleaning text: The raw data was cleaned by excluding unnecessary punctuation marks, special characters, redundant symbols and formatting irregularities. Newline characters were removed and additional white spaces were grayscale for uniformity of text. It

lessened noise and tokenized input quality was enhanced.

Text Normalization: To handle orthographic variations specific to Bangla, all text was normalized using the Bangla text normalizer developed by the CSE BUET NLP Group. The normalizer performs canonicalization of Bangla Unicode characters, standardizes punctuation usage, corrects inconsistent spacing, and maps text to a uniform representation. This normalization process was crucial to minimize token-level inconsistencies and enhance the tokenizer's effectiveness.

Tokenization: Articles and summaries were tokenized in the next step after normalization with Hugging Face's Auto Tokenizer corresponding to the pretrained models (BanglaT5 and mT5). Tokenization step has converted the Bangla sentences into token IDs and attention masks suitable for transformer model input. This step was critical to guarantee that all of the sequences are properly segmented into the model-readable components.

Padding and Truncation: Padding and truncation were both used to keep sequence length constant during training. For shorter sequences, they were padded with special tokens and for longer ones were trim to the maximum input length of the supported models. For the purpose of this study, the maximum length of input sequences and output sequences were set to 128-256 tokens respectively, taking into account the approximate length distributions observed in the dataset.

Data Partition: After preprocessing, data was further partitioned as 70% training, 20% validation, and 10% testing. This stratified splitting ensured that the models were trained on adequate amount of data while maintaining fair and not biased evaluation on unseen samples.

As such, these transformation steps were key to refine the raw BANS dataset and format it in a tidy and structured manner. They not only improved the effectiveness of the training but also lead to the positive performance of BanglaT5 and mT5 models when tested for the Bangla abstractive summarization task.

3. Model Selection

The application of BanglaT5 and mT5 models use to address the Bangla abstractive text summarization task. BanglaT5, being trained on a downstream Bengali-specific text, can efficiently capture the linguistic nuances embedded for Bangla consisting of it's rich morphology, syntactics and varying semantics. This model is great at producing short, on topic summaries of documents, and thus useful for abstractive summarization tasks. Alternatively, the mT5 model, a state-of-the-art multilingual transformer, makes a useful reference model. Trained on multiple languages LMs like mT5 can tackle summarization tasks not only in Bangla but in all the languages it was trained on. This multilingual evaluation allows us to determine the effectiveness of cross-lingual models for Bangla summarization alone and helps us understand the extent to which a universal model can work for low-resource languages such as Bangla.

By adopting this two-model approach, this work achieves a compromise between a language-specific model and a multilingual model, providing both a language specific tool for Bangla, and a strong general model for multilingual tasks. Despite this, models such as BERT have excelled on extractive summarization, but are nongenerative and thus do not fulfill the requirements for abstractive summarization. Unlike BERT, which extracts

passages from a fixed document, BanglaT5 and mT5 can generate summaries from the context of the input text, making them far more suited for generative text summarization tasks.

Together, these models provide a fine equilibrium between specialization and generality, contributing to the establishment of a strong baseline for advanced Bangla NLP tasks, including summarization. Both models follow the encoder-decoder architecture, as depicted in Figure 3.3.

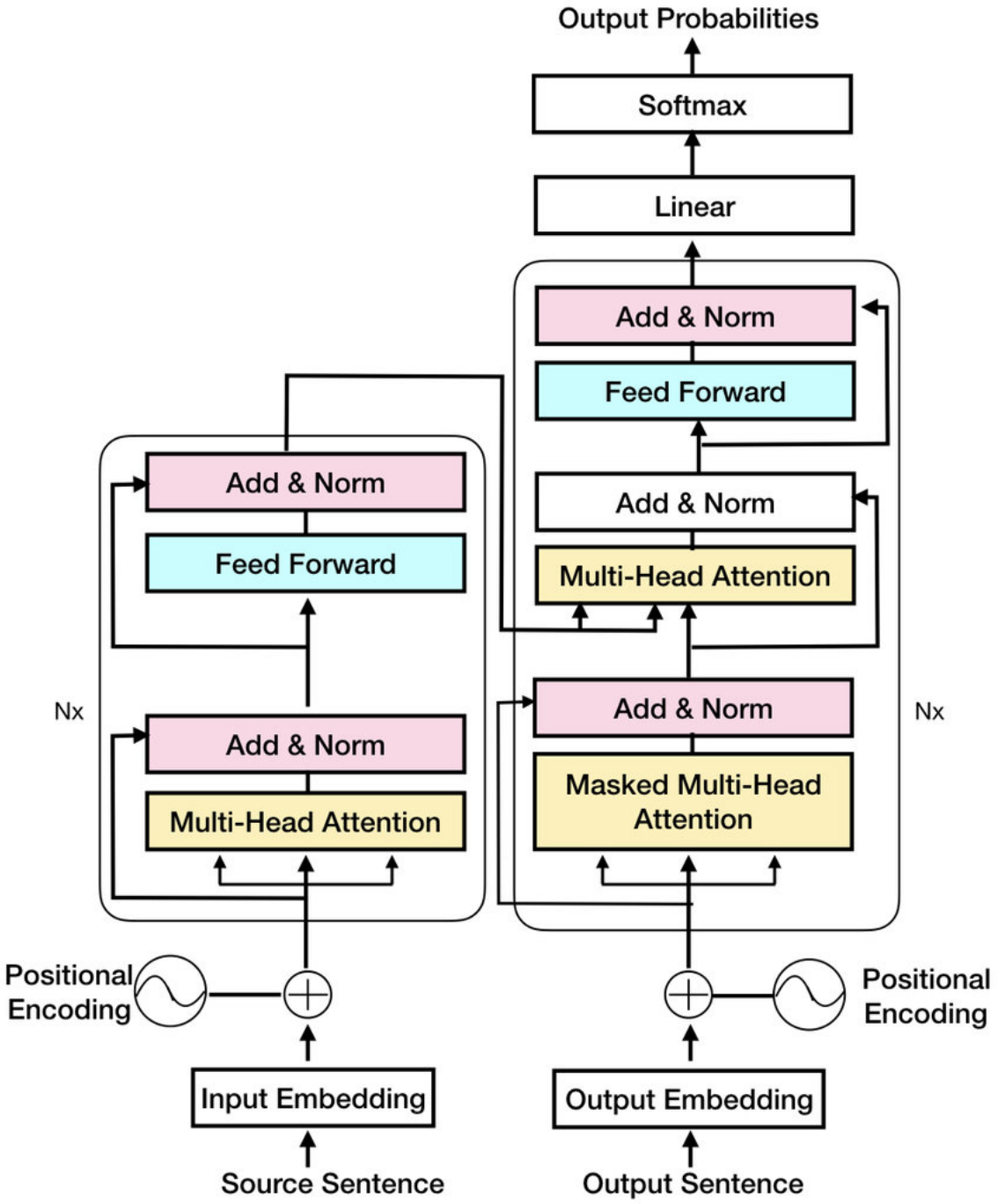


Figure 3.3: T5 Architecture

i. mT5: Multilingual T5 Model

The mT5 (Multilingual Text-to-Text Transfer Transformer) [21] model is a state-of-the-art transformer-based architecture that extends the successful T5 (Text-to-Text Transfer Transformer) model to multiple languages. Pre-trained on a vast range of languages, mT5 has been shown to perform effectively on a wide array of natural language processing (NLP) tasks by framing them as text-to-text problems. In this study, mT5 serves as a crucial multilingual model for comparing the performance of a specialized Bangla model (BanglaT5) with a generalized cross-lingual model.

Architecture: The mT5 architecture builds upon the original T5 model, which uses a denoising autoencoder objective to pre-train the model on a massive multilingual corpus. mT5 uses an encoder-decoder framework, where both the encoder and decoder consist of multiple layers of transformers, making it highly suitable for sequence-to-sequence tasks like text summarization, translation, and question answering. Unlike BERT (which is based solely on an encoder), mT5 can generate sequences, making it a generative model. mT5 consists of 12 layers in both the encoder and decoder, 12 attention heads, a 768 hidden size (base), or 512/1024 for small/large, and a feed-forward size of 2,048 (gated-GELU activation).

The mT5 model is pre-trained on over 100 languages using a text-to-text paradigm, meaning all tasks (summarization, translation, etc.) are approached as generating output text from input text. For each task, the model is fed with an input text (e.g., a news article for summarization) and expected to generate an output text (e.g., a summary). This flexibility allows mT5 to handle a wide range of NLP applications with a single architecture.

Pretraining and Multilingual Capabilities: mT5 is pre-trained on the C4 (Colossal Clean Crawled Corpus) dataset, which is a large-scale dataset extracted from web data across various languages. The model's multilingual ability comes from its training on text data in over 100 languages, enabling it to generalize across a diverse set of languages. By training on a huge multilingual corpus, mT5 is not just able to learn the syntax and semantics of several languages but also learns cross-lingual representations and can thus perform tasks in languages it was never explicitly trained for.

For Bangla summarization, mT5 gets an advantage from its ability to learn patterns not just in Bangla, but also across other Indic languages (such as Hindi, Tamil, etc.), which have linguistic similarities. This multi-lingual knowledge helps improve its summarization capabilities when applied to Bangla text.

ii. BanglaT5: Bangla-Specific T5 Model

BanglaT5[20] is a transformer-based model especially pre-trained for the Bangla language, following the same architecture as the original T5 (Text-to-Text Transfer Transformer) model. BanglaT5 is fine-tuned for various Bangla NLP tasks, with a primary focus on tasks such as summarization, translation, and question answering. The model is optimized to handle the linguistic intricacies of the Bangla language, including its rich morphology, syntactic structures, and semantics, which are often challenging for models not tailored to Bangla.

Architecture: Like T5, BanglaT5 employs an encoder-decoder architecture, where both the encoder and decoder consist of multi-layer transformers. The encoder reads the input text (in this case, Bangla articles), while the decoder generates the output (the corresponding summary in Bangla). The model operates in a text-to-text paradigm, meaning all tasks, including summarization, are framed as converting one text sequence (the article) into another (the summary). BanglaT5 consists of 12 layers in both the encoder and decoder, 12 attention heads, a hidden size of 768, and a feed-forward size of 2,048 (GeGLU activation). Training follows the T5 span-corruption objective.

The advantage of this encoder-decoder architecture is its ability to handle both extractive and generative tasks. While extractive models pick parts of the input text directly, generative models like BanglaT5 create new text from the input, making it more suitable for abstractive summarization. In this study, the model was fine-tuned specifically for Bangla abstractive summarization, where the goal is to produce fluent and concise summaries that are not direct extracts from the input text.

Pretraining and Fine-Tuning: BanglaT5 was pre-trained on a massive Bangla text corpus, which included several domains (news articles, blogs, and other publicly available Bangla resources). It was pre-trained on predicting the missing or corrupted words in the input sequence for learning the structure, syntax and semantic relationship of the Bangla text. This pre-training equips BanglaT5 with the capability to generalize to varied downstream tasks, specifically to tasks which demand deeper context understanding or text generation.

iii. GPT

In addition to BanglaT5 and mT5, we also explore the use of GPT-4o in Bangla abstractive summarization through prompt engineering. GPT-4o, a latest model in GPT (Generative Pre-trained Transformer) family, is based on a decoder-only transformer model. Unlike encoder-decoder models such as BanglaT5 and mT5, which take input and output independently, GPT-4o is designed with a single autoregressive transformer architecture. This enables the model to produce text in a step-by-step manner, i.e., token by token, based on the previous ones.

Architecture: The GPT model is a decoder-only transformer that autoregressively predicts the next word in a sequence of input tokens. Being autoregressive means, it generates output one step at a time, one token at a time, conditioned on the preceding tokens, which is more useful for applications such as text generation and summarization. Unlike the classic encoder-decoder models which decodes independently of encoding, GPT considers the task a text generation problem and formulates to be able to produce the contextually appropriate summaries from the text input. Model strengths Its strength lies in being able to produce fluent, human-like text, as well as the benefit of a single (but flexible) model architecture across a range of natural language processing tasks.

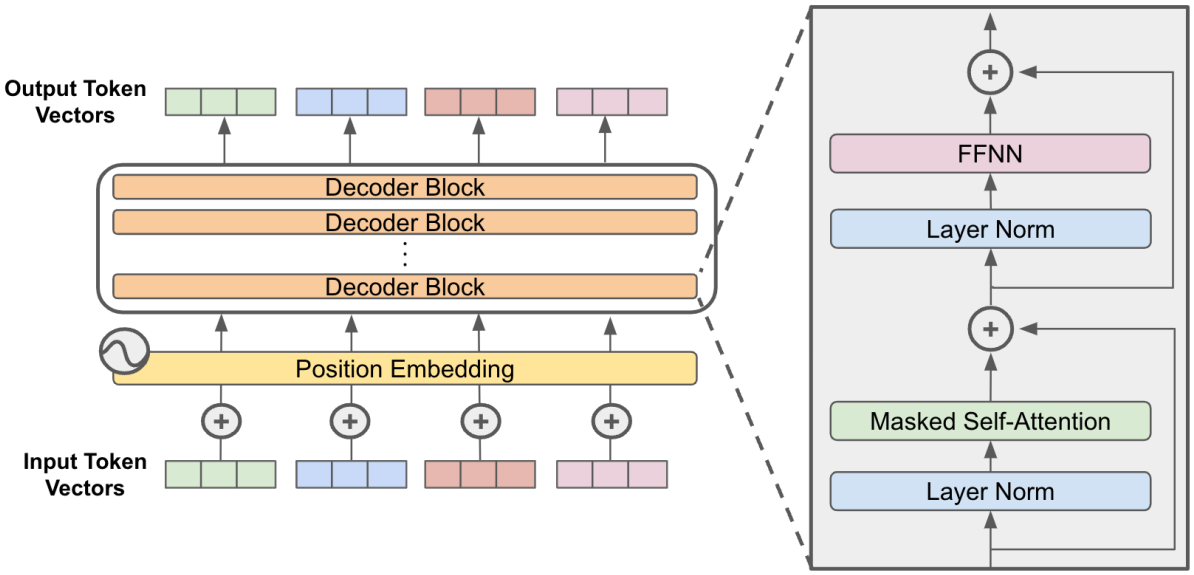


Figure 3.4: GPT Architecture

4. Model Training

In this research, we followed a modular training-based approach to fine-tune two separate models BanglaT5, mT5 on Bangla abstractive summarization. We fine-tuned each model individually to better suit the specific needs of the summarization task. The fine-turning was performed for the details of Bangla language, trained on the BANS dataset. This enabled us to benefit from the language-specific and multilingual transformer models and prompted-engineered generative models (GPT-40) capabilities. The training included extensive tuning of hyperparameters and task specific loss functions and exhaustive use of state-of-the-art training methods to achieve high quality while ensuring the model was capable of generating fluent, contextually appropriate and factually grounded summaries.

i. BanglaT5

BanglaT5 fine-tuned model was further trained for Bangla abstractive summarization using the BANS dataset. Fine-tuning was done using the AdamW optimizer with a weight decay of 0.01 and gradient accumulation for better memory efficiency. The training was done for 15 epochs with a batch size of 20 to balance between computation efficiency and training effectiveness. Efforts were also made to accelerate the training process while keeping memory overhead as low as possible.

Cross-entropy loss was the main goal of the training to make a model which produces summaries that are true according to the context. The model was built in a structure to model the intricate syntactic and morphological characteristics of the Bangla language, making it optimal for Bangla abstractive summarization. Table 3.4 shows the major training parameters and hyperparameters used during the fine-tuning of the BanglaT5 model.

Table 3.4: Training Arguments for BanglaT5

Parameter	Value
Model	BanglaT5
Loss Function	Cross-entropy loss
Epochs	15
Batch Size	20
Optimizer	AdamW
Learning Rate	1e-3
Weight Decay	0.01
Gradient Accumulation	8
Max Length (Input/Output)	128
Warmup Steps	100
Learning Rate Scheduler	cosine_with_restarts
Evaluation Strategy	steps

ii. mT5 Model

The mT5 model, a multilingual version of T5 pre-trained across multiple languages, was fine-tuned for Bangla abstractive summarization using the BANS dataset. The fine-tuning process uses the AdamW optimizer with a weight decay of 0.01 and included gradient accumulation to optimize memory usage and training efficiency. The model was trained for 15 epochs with a batch size of 4 to maintain a balance between computational efficiency and effective learning.

The training objective for mT5 was to minimize cross-entropy loss, ensuring that the model could generate fluent, concise, and contextually accurate summaries. While mT5 was pre-trained on a multilingual corpus, the fine-tuning process focused on enhancing its ability to capture the linguistic and syntactical features of Bangla for summarization tasks. The model's architecture, based on the encoder-decoder transformer design, was suited to handle the complex syntactic and semantic nuances of Bangla text. Table 3.2 provides a summary of the key training arguments and hyperparameters used for fine-tuning the mT5 model.

Table 3.5: Training Arguments for mT5

Parameter	Value
Model	mT5
Loss Function	Cross-entropy loss

Epochs	15
Batch Size	4
Optimizer	AdamW
Learning Rate	1e-3
Weight Decay	0.01
Gradient Accumulation	8
Max Length (Input/Output)	128
Warmup Steps	100
Learning Rate Scheduler	cosine_with_restarts
Evaluation Strategy	epoch

iii. GPT

In addition to fine-tuning models like BanglaT5 and mT5, ChatGPT was employed for Bangla abstractive summarization through prompt engineering. Rather than fine-tuning the model on a specific dataset, ChatGPT was utilized in a zero-shot or few-shot setting, where the model was provided with prompts and asked to generate summaries based on the Bangla text.

The prompt engineering approach for ChatGPT involved designing explicit and clear instructions to guide the model in generating contextually relevant and concise summaries of Bangla articles. Table 3.2 provides a summary of instruction used for the GPT model.

Table 3.6: Training Arguments for GPT

Parameter	Value
Model	GPT-4o
Training Type	Zero-shot
Prompt	I will give you Bangla text.you have to give me an abstract bangla summary for that text.

5. Model Evaluation

To evaluate the performance of BanglaT5 and mT5 on the Bangla abstractive summarization task, we used BERTScore [23], a metric that leverages BERT embeddings to compute the similarity between the predicted summary and the reference summary. BERTScore evaluates the quality of generated text by comparing the similarity of token embeddings rather than using traditional n-gram overlaps.

i. BERTScore Precision

Precision measures how many of the tokens in the predicted summary match those in the reference summary. It is defined as:

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\max_{j=1}^{M}\text{sim}(p_i, r_j)$$

- p_i is the embedding of the predicted token at position i,
- r_j is the embedding of the reference token at position j,
- sim represents cosine similarity between the embeddings of p_i and r_j,
- N is the number of tokens in the predicted summary,
- M is the number of tokens in the reference summary.

ii. BERTScore Recall

Recall measures how many of the tokens in the reference summary are captured by the predicted summary. It is defined as:

$$\text{Recall} = \frac{1}{M}\sum_{j=1}^{M}\max_{i=1}^{N}\text{sim}(p_i, r_j)$$

- p_i is the embedding of the predicted token at position i,
- r_j is the embedding of the reference token at position j,
- sim represents cosine similarity,
- N is the number of tokens in the predicted summary,
- M is the number of tokens in the reference summary.

iii. BERTScore F1-Score

The F1-Score is the harmonic mean of Precision and Recall, providing a balanced measure of performance. It is calculated as:

$$F1\,Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

iv. Human Evaluation

To evaluate the quality of the summaries generated by BanglaT5 and GPT, we conducted human evaluation by Bangla department professors (Dr. Rezwana Abedin & MD. Shamim Hossain) of Jahangirnagar University. Relevance, Coherence, and Conciseness. These criteria were selected to guarantee that the summary texts are grammatically correct and that there was capture the essential meaning from the original articles in a concise, relevant and effective way.

i. Relevance

Relevance checks how much of the core content and the key ideas of the original article are captured in the generated summary. A relevant summary should accurately convey the key points of the article while avoiding unnecessary or off-topic information.

- High Relevance: A summary that reflects the main content or findings presented in the article without including any unrelated or irrelevant information.
- Low Relevance: Summary contains information unrelated to the article or omit details necessary for understanding the main ideas in the original article

Evaluation scale (1-10):

- 10: Highly relevant, captures all essential points from the article.
- 7-9: Mostly relevant, but some minor details are missed or irrelevant points are included.
- 4-6: Moderately relevant, misses several important points or includes unnecessary details.
- 1-3: Irrelevant, misses key points and introduces off-topic information.

ii. Coherence

Coherence evaluates the logical flow and readability of the generated summary. A coherent summary should have a clear structure, where sentences connect smoothly and ideas are presented in a logical order.

- High Coherence: A summary that is well-structured, with sentences that flow naturally from one to the next. The summary maintains consistency in style, tone, and meaning.
- Low Coherence: A summary that feels disjointed or fragmented, where ideas do not flow logically.

Evaluation scale (1-10):

- 10: Very coherent, easy to follow with clear logical flow.
- 7-9: Mostly coherent, but may have minor issues with sentence flow or logical connections.
- 4-6: Somewhat coherent, difficult to follow at times.
- 1-3: Poor coherence, hard to follow and disjointed.

iii. Conciseness

Conciseness measures how well the model condenses the article into a brief summary without losing critical information. The summary should be short yet comprehensive, avoiding unnecessary details or repetition.

- High Conciseness: A summary that is brief, containing only the essential information from the article. The summary is compact and focused, omitting redundant or irrelevant details.

- Low Conciseness: A summary that is either too long, including unnecessary information, or too short, leaving out important aspects of the article.

Evaluation scale (1-10):

- 10: Highly concise, captures all key points in a brief and focused manner.
- 7-9: Moderately concise, but with some unnecessary information or minor omissions.
- 4-6: Not concise, either too verbose or missing important information.
- 1-3: Too long or too brief, with significant information missing or redundant.

3.3 UI Design

To ensure accessibility and usability of the Bangla abstractive summarization system, a simple and intuitive user interface (UI) was developed. The user interface was built using Flask for offline and Gradio for online, which provides an interactive front-end for our models and allows seamless integration with the backend summarization pipeline.

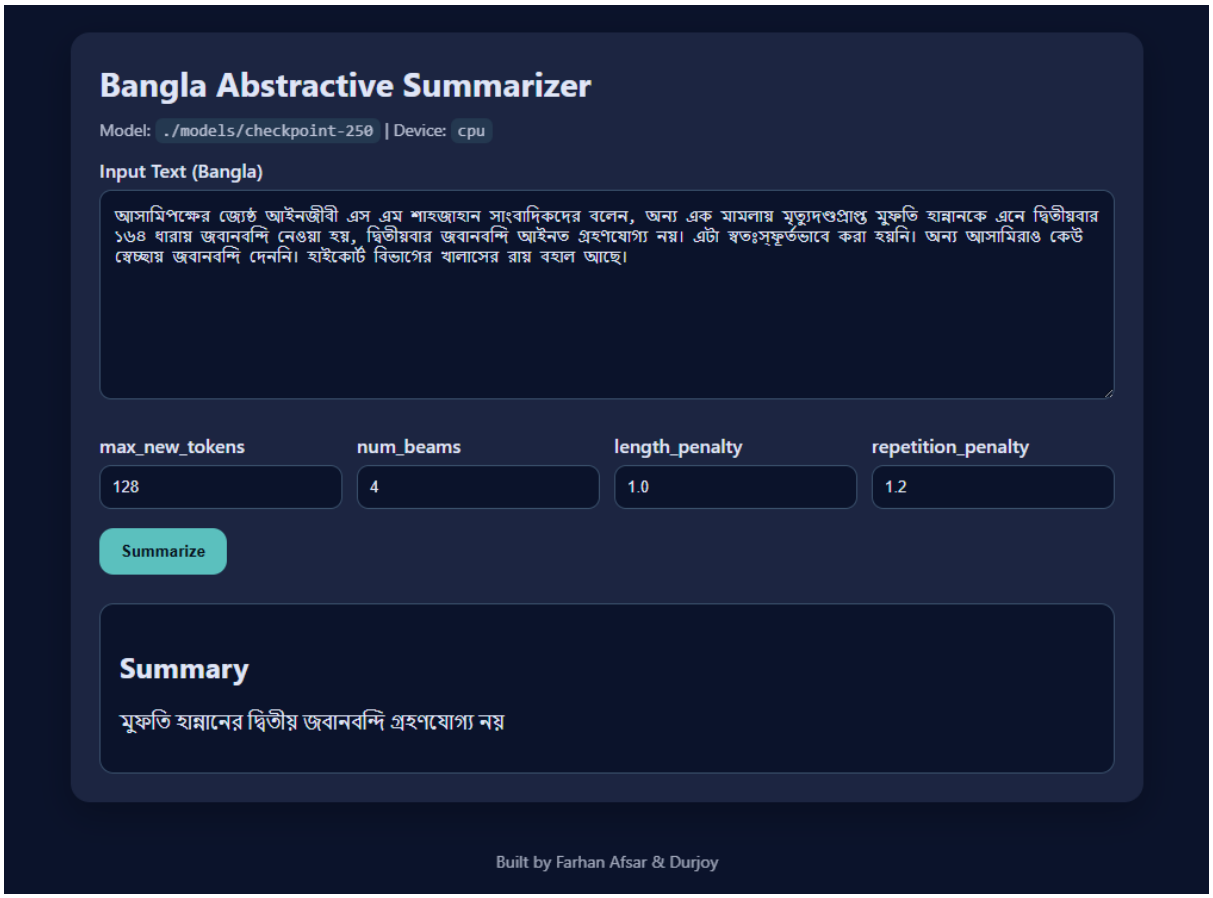


Figure 3.5: User Interface for web application

Users can provide Bangla text in the dedicated input box and set the summarization parameters through the UI shown in Figure 3.5. There are four adjustable hyperparameters which includes max_new_tokens, num_beams, length_penalty and repetition_penalty. They enable the user to adjust the size of the summaries to be created, how different outputs ought to be from each other, and to punish repetition in the output. Just press the one and only Summarize button to start the output which you will find in the Summary section below.

The user interface design promises clarity, simplicity, and control. This means a clean, organized layout where each field is capable of being labelled, so that any user can understand and interact with the system, regardless of their technical expertise. For clarity, at the top of the generated image display the model details (check point, device used, etc.). It is a lightweight and responsive UI which can be deploy on places like Hugging-face Spaces, widening the access to researchers and general users.

In short, the user interface is a link that connects the abstract abstractive summarization model with people and makes the model a real-life implementational tool that can be used.

3.4 Project Plan

The project plan outlines a clear and structured roadmap, detailing the methodology and evaluation plans for developing BanglaT5, mT5, and prompt engineered GPT for the task of Bangla abstractive summarization It provides us a map that details of every major task to be done, including data preparation, building, training and evaluation, and then reporting. Phases will be completed on time lines that help to prevent reworking of projects.

Gantt chart at Figure 3.6 visually presents the project timeline: Project planning has been finalized and will follow by dataset preparation, model development, training and then finally evaluation, documentation and reporting. This shows project objectives with the help of this organized timeline and in a sequential format. Each phase has a specific time frame and corresponding deliverables to facilitate control of resources and the completion of tasks within a specified period.

The timeline shows the phases and tasks for the project plan:

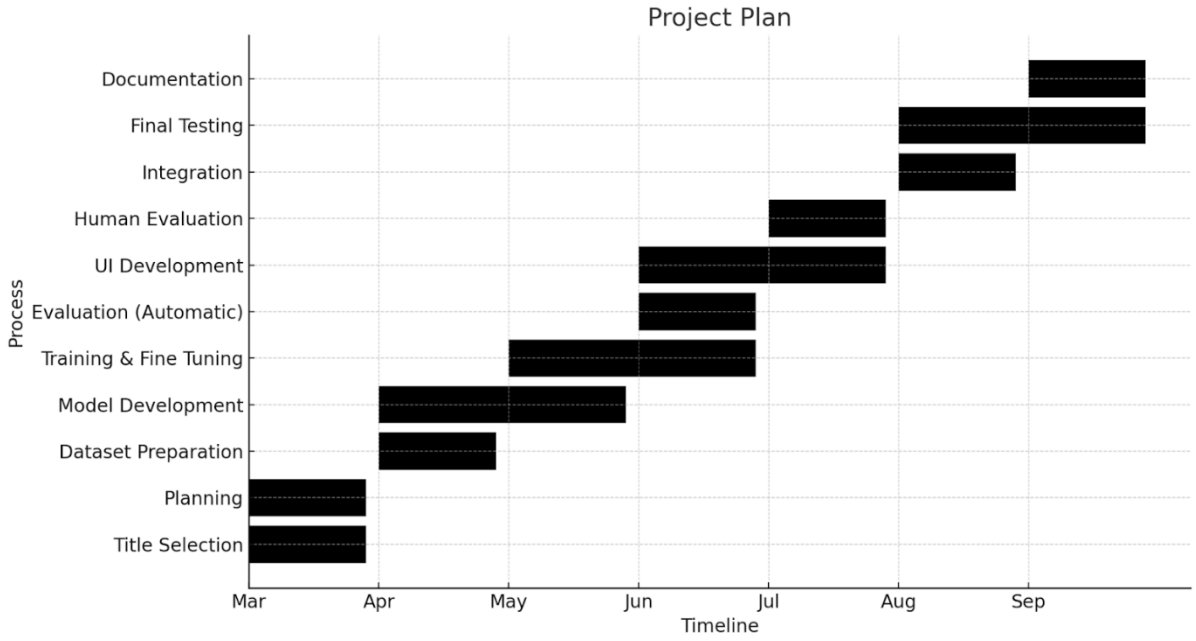


Figure 3.6: Project Timeline Gantt Chart

3.5 Task Allocation

Effective task allocation is vital for the smooth execution of the Bangla Abstractive Summarization project. The team is organized into distinct roles based on individual expertise to ensure a seamless workflow and collaboration throughout the project.

Table 3.7: Task Allocation Table

Task	Md. Farhan Afsar	Naimur Rahman Durjoy
Dataset Collection and Preparation	✓	✓

Model Development	✓	×
Model FIne Tuning	✓	×
Model Evaluation	✓	×
Human Evaluation	×	✓
Literature Review	×	✓
UI Development	✓	✓
Documentation & Reporting	✓	✓

3.6 Summary

In this chapter, we discussed the overall methodology of the study — from gathering and preprocessing the BANS dataset (which was cleaned, normalized, and tokenized, in addition to student GG data being partitioned to maintain linguistic consistency and prepare it for model training). We used BanglaT5 and mT5 for models fine-tuned on the dataset and a GPT model used through prompt engineering to provide a zero-shot comparison. We briefly describe the training strategy (AdamW optimizer, learning rate scheduling, gradient accumulation, task-specific hyperparameters) and evaluation (automatic metrics like BERTScore and human evaluation on the dimensions of relevance, coherence, and conciseness). The task of execution was described clearly as a hierarchical Gantt chart format outlined a project plan and where each task was assigned. As a whole, the methodology synthesized data preparation, model training, evaluation and planning using a coherent framework that provided a solid basis for the following results and discussion.

Chapter 4

Implementation and Results

This chapter presents the implementation details of the BanglaT5 and mT5 models, including environment setup, training procedures, and evaluation. It discusses both automatic metrics and human evaluation results, supported by charts, curves, and detailed analysis.

4.1 Environment Setup

To implement and evaluate the Bangla abstractive text summarization models, Python 3.12 was chosen for its stability and availability of most of the state-of-art NLP libraries. We performed our experiments mainly on Kaggle using NVIDIA Tesla T4 GPUs which are enough to fine-tune such large-scale transformer models (BanglaT5 and mT5) having huge memory requirement. Also, MacBook Air M1 and Windows desktop machine with Intel i5-12400 and 16 GB RAM are used for local experiments and preprocessing, thus allowing cloud and local development.

The deep learning pipeline was built on PyTorch and the Hugging Face Transformers library, which enabled efficient model loading, tokenization, and fine-tuning. Dataset preprocessing and manipulation were carried out using Pandas and NumPy, while evaluation leveraged Scikit-learn and BERTScore for automatic metric-based assessment. Visualization of training dynamics and evaluation outputs was supported through Matplotlib and Seaborn. For deployment, Gradio was used to build an interactive interface, and models were hosted on Hugging Face Spaces to enable web-based access.

This environment ensured smooth experimentation, efficient GPU utilization, and robust support for Bangla language processing. The detailed setup is presented in Table 4.1.

Table 4.1: System Environment Configuration

Component	Details
Programming Language	Python 3.12
Development Platforms	Kaggle Notebook (NVIDIA Tesla T4 GPU), MacBook Air M1, Windows i5-12400 (16 GB RAM)
Deep Learning Stack	PyTorch, Hugging Face Transformers
Evaluation Libraries	BERTScore
Data Handling	Pandas, NumPy
Visualization Tools	Matplotlib, Seaborn
Deployment Tools	Gradio (UI), Hugging Face Spaces (Model Hosting)
Models Used	BanglaT5, mT5, GPT (prompt-engineered for comparison)

4.2 Comparative Analysis

To assess the effectiveness and generalization of the Bangla abstractive summarization systems, we evaluated (i) fine-tuned encoder–decoder models (BanglaT5, mT5) on the BANS dataset and (ii) a prompt-engineered GPT baseline. Evaluation combined automatic metrics (BERTScore) and human judgments (relevance, coherence, conciseness), enabling a balanced comparison between semantic overlap and perceived quality.

i. Multilingual T5:

To evaluate the performance of the multilingual mT5 model on the Bangla Abstractive News Summarization (BANS) dataset, the model was fine-tuned for 15 epochs with task-specific hyperparameters. As a multilingual encoder–decoder transformer, mT5 was expected to capture cross-lingual representations; however, its adaptation to Bangla required careful observation of training dynamics and evaluation metrics. The following subsections present the training and validation loss curves across epochs and the automatic evaluation results using BERTScore.

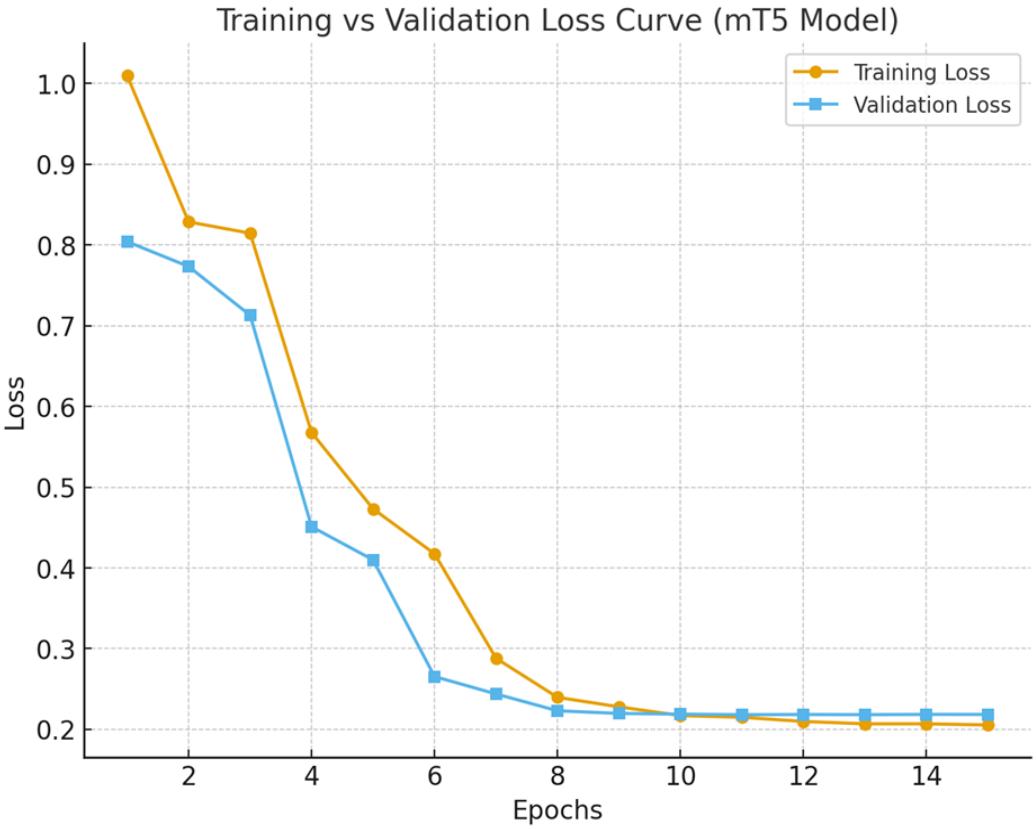


Figure 4.1: Training vs Validation Loss of mT5 Fine Tuning

The training and validation loss curves for the mT5 model over 15 epochs are shown in Figure 4.1. At the beginning of training (epoch 1), the training loss was 1.0098 and the validation loss was 0.8041, reflecting the difficulty of initializing a large multilingual model for Bangla summarization. Over successive epochs, both curves exhibited a sharp downward trend, particularly between epochs 3 and 6, where the training loss decreased from 0.8146 to 0.2881, and the validation loss from 0.7129 to 0.2439. This phase indicates that the model was rapidly learning useful representations of the dataset.

From epoch 7 onwards, both training and validation loss values stabilized, converging around 0.21–0.22 by epoch 15. The convergence of the two curves without significant divergence suggests that the model generalized well to unseen data, with no strong indication of overfitting. The narrow gap between training and validation loss after epoch 10 further validates that the optimization was stable and efficient. This steady decline, followed by stabilization, confirms the effectiveness of fine-tuning mT5 on the BANS dataset.

Automatic Evaluation with BERTScore

The performance of the mT5 model was further assessed using BERTScore, evaluated under two encoder configurations: Bangla-aligned (lang="bn") and cross-lingual (lang="en"). Results are summarized in Table 4.2.

Table 4.2: BERTScore of mT5

Setting	Precision	Recall	F1
Lang=”bn”	0.5389	0.5634	0.5507
Lang="en"	0.7885	0.7445	0.7651

The results show that mT5 achieved a relatively low BERTScore F1 of 0.5507 when evaluated with the Bangla encoder, indicating that the model struggled to fully capture Bangla-specific semantics. However, the F1 score improved to 0.7651 under the English encoder. This improvement can be attributed to the fact that BERTScore with lang="en" relies on large cross-lingual encoders such as XLM-RoBERTa, which have been extensively pre-trained on massive multilingual corpora. These encoders often have richer subword coverage and more robust semantic embeddings than the Bangla-specific setting, where the underlying pretrained space is comparatively limited. As a result, the English configuration provides more stable similarity judgments across languages, indirectly boosting the evaluation scores. While this highlights the benefit of mT5’s multilingual pretraining, it also reflects a limitation in Bangla-specific adaptation, since a truly Bangla-optimized model should not depend on English-aligned evaluation encoders to demonstrate strong performance.

ii. BanglaT5

BanglaT5, a language-specific encoder–decoder transformer pre-trained exclusively on Bangla corpora, was fine-tuned on the BANS dataset for abstractive summarization. Owing to its domain-focused pretraining, the model was expected to outperform general multilingual models by capturing the syntactic and semantic richness of Bangla more effectively. The following subsections present the training and validation loss behavior across epochs and the automatic evaluation results obtained using BERTScore.

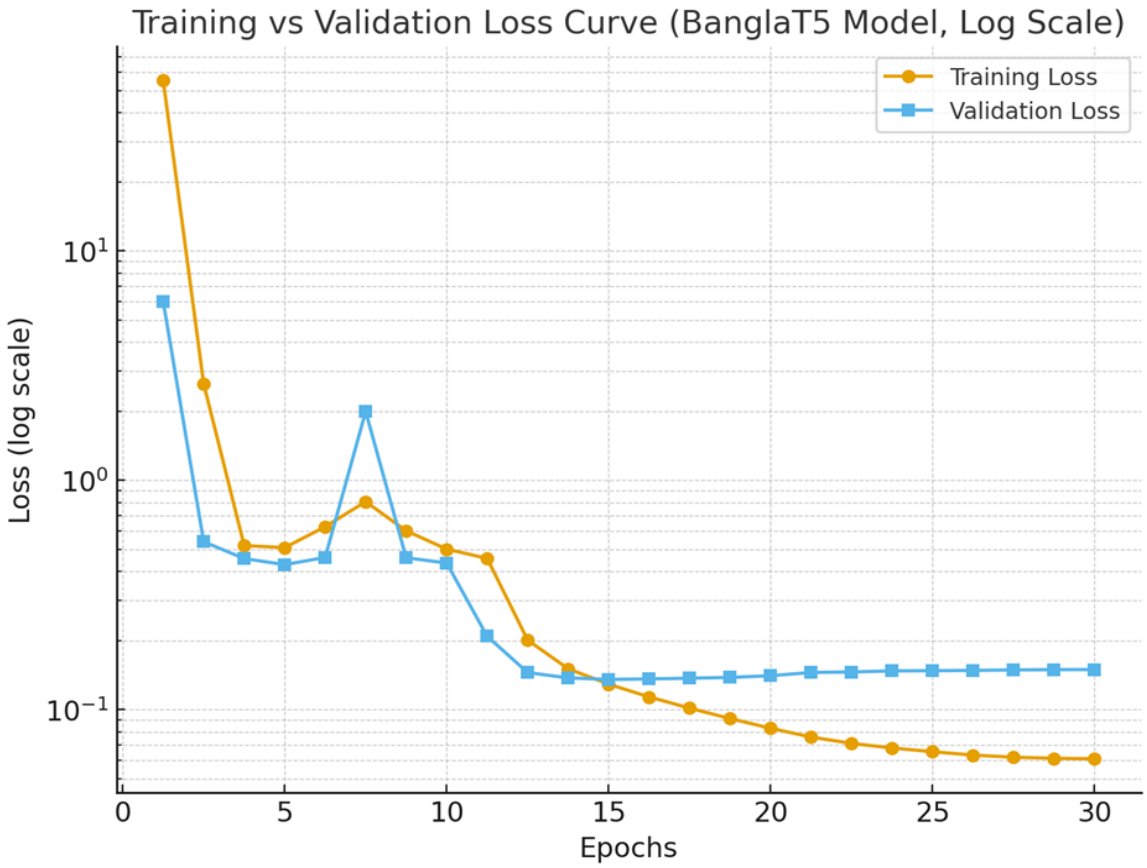


Figure 4.2: Training vs Validation Loss of BanglaT5 Fine Tuning

The training and validation loss curves for BanglaT5 over 30 epochs are illustrated in Figure 4.X. At the beginning of training (epoch 1), the model exhibited a very high training loss (55.36) and validation loss (6.02), which is typical during the early optimization phase. However, by epoch 3, both metrics dropped sharply, with training loss falling below 1.0 and validation loss stabilizing around 0.45, reflecting rapid adaptation to the summarization task.

From epochs 5 to 15, both curves showed steady improvement: training loss decreased to approximately 0.20, while validation loss reached 0.14, demonstrating strong generalization. After epoch 15, the training loss continued to decline gradually, reaching 0.061 by epoch 30, whereas the validation loss plateaued within the 0.135–0.15 range. The convergence of the two curves without major divergence indicates stable optimization and an absence of significant overfitting.

Automatic Evaluation with BERTScore

BanglaT5 was further evaluated using BERTScore to measure semantic similarity between generated and gold summaries. Results were computed using both Bangla-specific (lang="bn") and multilingual (lang="en") encoders, as shown in Table 4.3

Table 4.3: BERTScore of BanglaT5

Setting	Precision	Recall	F1
Lang=”bn”	0.8155	0.8204	0.8174
Lang="en"	0.9562	0.9677	0.9569

The Bangla-specific evaluation yielded a strong F1 score of 0.8174, confirming the model's ability to capture semantic fidelity in the native language. Under the English configuration, the F1 increased to 0.9569. This rise is primarily due to the richer subword coverage and robust semantic embeddings of multilingual encoders such as XLM-RoBERTa, which underpin the lang="en" setting. While this boosts evaluation scores, it also highlights a known bias where multilingual evaluation models inflate performance compared to Bangla-native alignment.

Comparative Analysis: BanglaT5 vs mT5

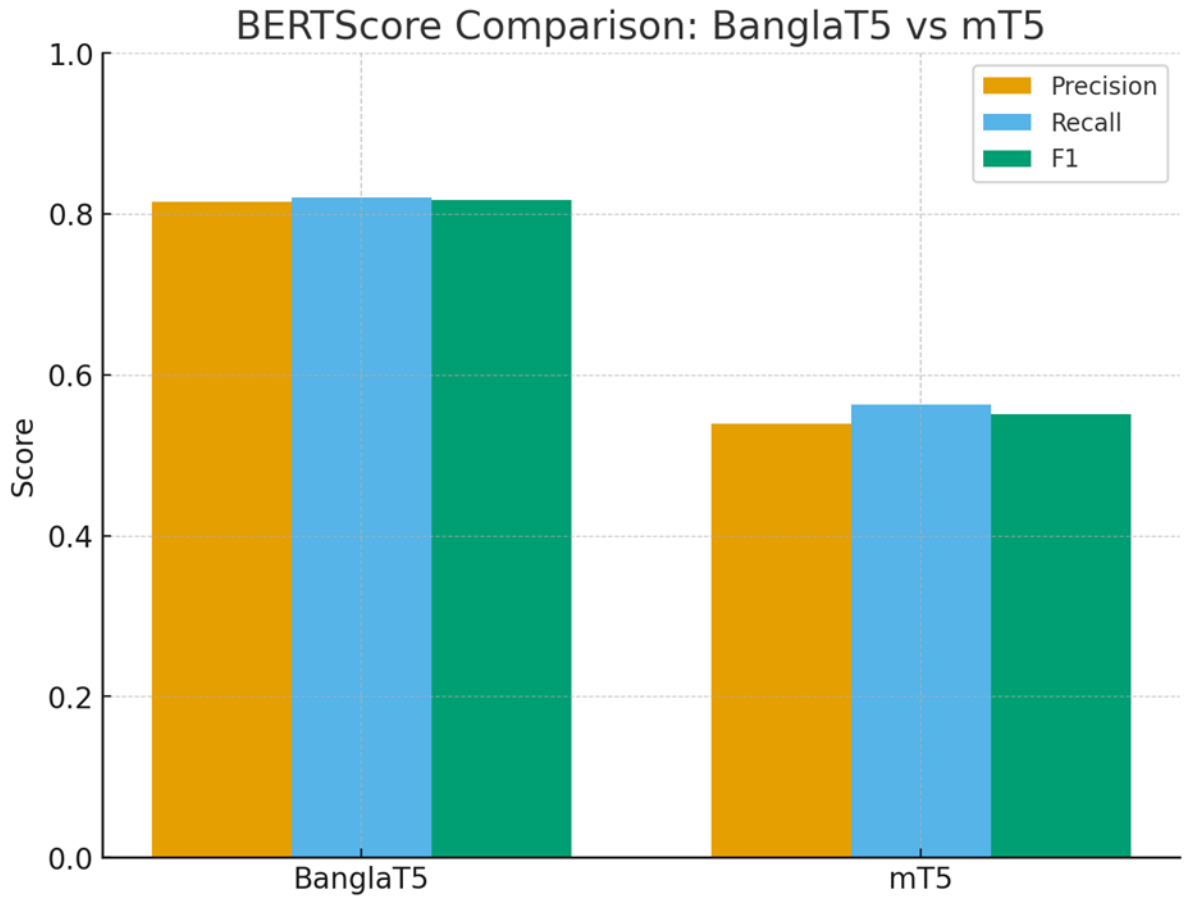


Figure 4.3: BERTScore Comparison mT5 vs BanglaT5

The results clearly establish that BanglaT5 substantially outperforms mT5 across all metrics. BanglaT5 achieved an F1 score of 0.8174, which is almost 50% higher than mT5's 0.5507. Precision and recall show a similar trend: BanglaT5 maintains balanced values above 0.81, while mT5 lags at 0.54–0.56. This indicates that BanglaT5 not only generates summaries with greater semantic fidelity but also captures more of the relevant content consistently.

Human Evaluation

To complement automatic metrics, a human evaluation study was conducted using three independent annotators. Each annotator rated 50 summaries per system (GPT and BanglaT5) on three quality dimensions: Relevance, Coherence, and Conciseness. Ratings were given on a 0–10 scale, which were aggregated (maximum = 500 per annotator per dimension) and normalized into percentages for comparability.

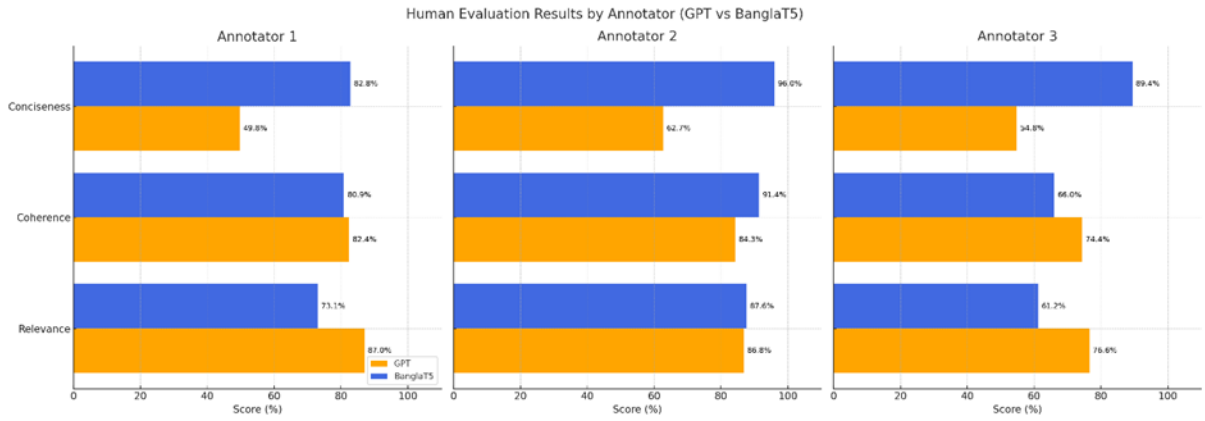


Figure 4.4: Annotator Evaluation Result

Three annotators independently evaluated 50 summaries per system on a 0–10 scale across three dimensions: Relevance, Coherence, and Conciseness. The normalized percentages for each annotator are presented in Figure 4.4.

Annotator 1 rated GPT higher in Relevance (87.0%) and Coherence (82.4%) but much lower in Conciseness (49.8%). In contrast, BanglaT5 scored lower in Relevance (73.1%) but performed competitively in Coherence (80.9%) and significantly better in Conciseness (82.8%).

Annotator 2 gave both systems strong scores but with different emphases: GPT achieved 86.8% in Relevance and 84.3% in Coherence, while BanglaT5 surpassed GPT in all three dimensions, particularly Conciseness (96.0%) and Coherence (91.4%).

Annotator 3 rated GPT consistently higher for Relevance (76.6%) and Coherence (74.4%), whereas BanglaT5 again dominated in Conciseness (89.4%) but lagged behind in the other two dimensions (Relevance 61.2%, Coherence 66.0%).

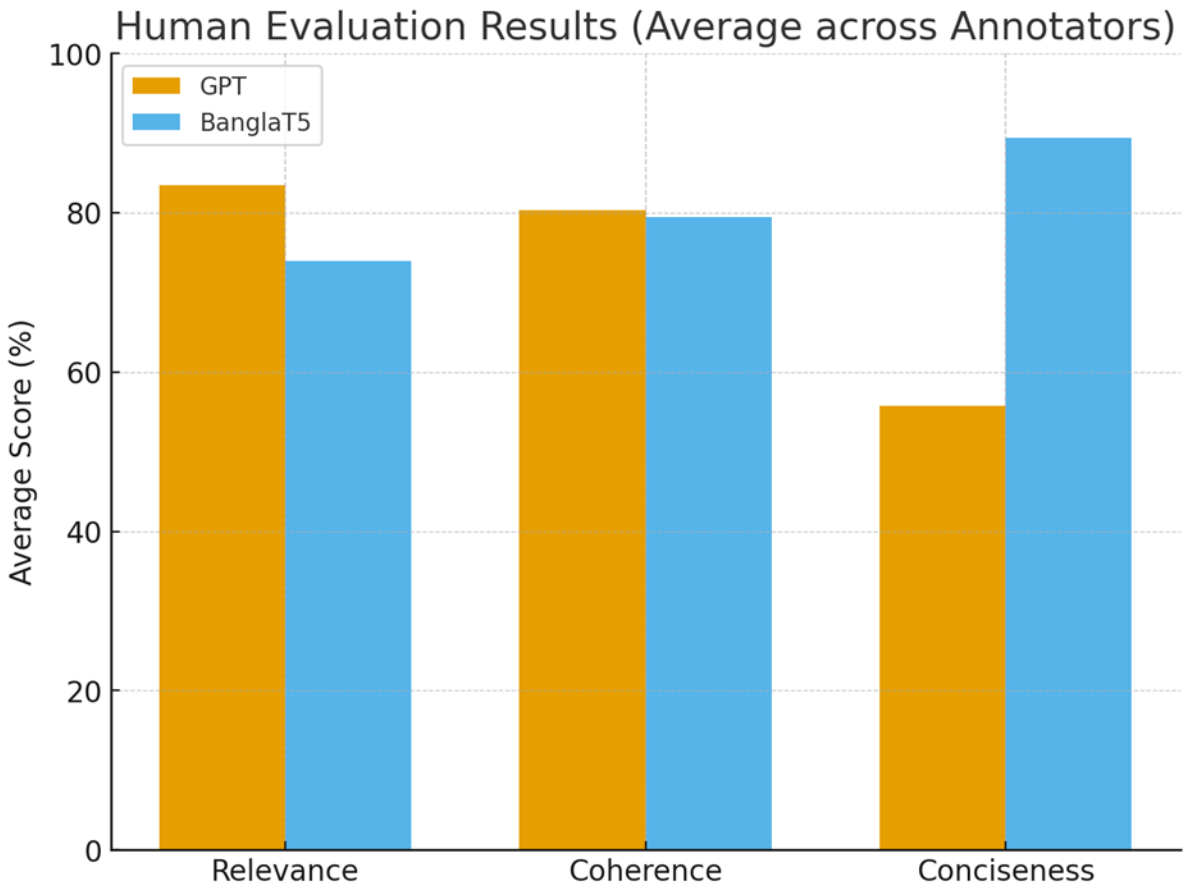


Figure 4.4: Annotator Average Evaluation Result

Averaging across annotators, GPT scored 83.5% in Relevance, 80.4% in Coherence, and 55.8% in Conciseness, while BanglaT5 scored 74.0% in Relevance, 79.4% in Coherence, and 89.4% in Conciseness. These results highlight a consistent pattern: GPT is perceived as more relevant and coherent overall, producing fluent and content-rich summaries, but tends to be verbose, leading to low conciseness ratings. Conversely, BanglaT5 generates shorter, headline-style summaries that better satisfy conciseness requirements but occasionally sacrifice depth of content coverage.

Overall, the human evaluation underscores a clear trade-off between the two approaches: GPT excels in content preservation and fluency, whereas BanglaT5 excels in brevity and compression. This suggests that the choice of model depends on application needs—BanglaT5 for contexts requiring concise summaries (e.g., news headlines) and GPT for contexts where completeness and readability are prioritized.

4.3 Results and Discussion

The experimental evaluation of BanglaT5, mT5, and GPT was carried out using both automatic metrics (BERTScore) and human judgments.

- BanglaT5 achieved an F1 of 0.8174 under the Bangla evaluation setting, and 0.9569 under the English evaluation setting. Its human evaluation results showed strong Conciseness (89.4%), with slightly lower Relevance (74.0%) and Coherence (79.4%).

- mT5 obtained significantly lower scores, with an F1 of 0.5507 (lang="bn") and 0.7651 (lang="en"). Human evaluation was not conducted for mT5 due to its weaker baseline results.

- GPT (prompt-engineered) was not fine-tuned but produced coherent outputs via prompting. Human evaluation results indicated higher Relevance (83.5%) and Coherence (80.4%), though Conciseness was much lower (55.8%) compared to BanglaT5.

The evaluation results provide several important insights into the relative strengths and weaknesses of BanglaT5, mT5, and GPT for Bangla abstractive summarization.

First, BanglaT5 consistently outperformed mT5 in both automatic and human evaluations. Its language-specific pretraining on Bangla corpora enabled stronger handling of Bangla morphology and syntax, leading to an F1 score of 0.8174 under Bangla-specific BERTScore evaluation. In human evaluation, BanglaT5 was rated extremely high in Conciseness (89.4%), showing its ability to generate short, headline-style summaries aligned with the style of reference summaries. However, its Relevance (74.0%) was lower than GPT, indicating that in compressing content, BanglaT5 occasionally omitted key details.

Second, GPT demonstrated superior Relevance (83.5%) and Coherence (80.4%), which can be attributed to its massive pretraining on diverse multilingual corpora and its generative flexibility. GPT outputs were judged to capture the main ideas more completely and to be more fluent overall. The disadvantage, however, was its tendency toward verbosity, reflected in a much lower Conciseness score (55.8%). This makes GPT less suited for tasks requiring short, headline-like outputs, but effective for contexts where readability and detail are prioritized.

Third, mT5 lagged behind both BanglaT5 and GPT. Its multilingual training, while offering broad cross-lingual capabilities, diluted its ability to specialize in Bangla, resulting in a relatively poor F1 score (0.5507) under Bangla evaluation. Without additional domain-adaptive pretraining, mT5 is less competitive for Bangla summarization tasks and serves mainly as a baseline.

The trade-offs across models are therefore clear:

- BanglaT5 → Strength: Concise, headline-style summaries; Weakness: risk of missing fine-grained details.

- GPT → Strength: Fluent, relevant, coherent summaries; Weakness: verbosity and lack of strict length control.

- mT5 → Strength: multilingual adaptability; Weakness: underperformance in Bangla-specific tasks.

From an application perspective, these findings suggest that BanglaT5 is best suited for news summarization systems requiring short, direct summaries, while GPT can serve as a valuable zero-shot generator where fluency and coverage are prioritized. Combining the two approaches—for example, generating with GPT and compressing with BanglaT5 could yield a hybrid system that balances conciseness with fluency.

Finally, there remains scope for further improvement. For BanglaT5, reinforcement learning with human feedback (RLHF) or controlled decoding strategies could help improve Relevance while preserving conciseness. For GPT, post-processing or constraint-based generation could reduce verbosity. For mT5, targeted pretraining on Bangla corpora would likely boost its performance

4.4 Summary

This chapter presented the complete implementation and evaluation of the proposed Bangla abstractive text summarization systems. The experimental environment was first described, covering both local and cloud-based setups using Python, PyTorch, Hugging Face Transformers, and GPU resources. Three models were evaluated: BanglaT5 (fine-tuned language-specific model), mT5 (fine-tuned multilingual baseline), and GPT (prompt-engineered without fine-tuning).

The training and validation loss curves confirmed stable convergence for both BanglaT5 and mT5, with BanglaT5 achieving lower final loss values and demonstrating efficient optimization without overfitting. Automatic evaluation using BERTScore highlighted the superiority of BanglaT5 (F1 = 0.8174 under Bangla evaluation) over mT5 (F1 = 0.5507), while GPT, though not directly comparable on this metric, showed strong human-perceived quality.

The human evaluation provided further insights. GPT received higher ratings for Relevance (83.5%) and Coherence (80.4%), reflecting its fluency and ability to preserve key ideas. BanglaT5, however, excelled in Conciseness (89.4%), producing short, headline-style summaries more aligned with abstractive news writing. Annotator-wise analysis confirmed these trends across all three evaluators.

Example summaries further illustrated the differences: GPT generated fluent but verbose outputs, while BanglaT5 produced compact summaries at times omitting minor details. Comparative analysis revealed the trade-offs among the models: BanglaT5 is well-suited for applications requiring brevity, GPT is preferable for fluency and completeness, and mT5 remains a weaker baseline without Bangla-specific adaptation.

In conclusion, this chapter established that BanglaT5 is the most effective model for Bangla abstractive summarization, balancing semantic fidelity with brevity, while GPT offers complementary strengths in fluency and relevance. The discussion also highlighted potential improvements, including reinforcement learning with human feedback, controlled decoding, and further pretraining for multilingual models. These results form the foundation for the engineering standards, challenges, and conclusions addressed in the next chapters.

Chapter 5

Engineering Standards and Design Challenges

This chapter maps the project against complex engineering problem-solving standards, knowledge profiles, and engineering activities. It also discusses design constraints, communication standards, ethical aspects, sustainability plans, and budget considerations.

This Chapter discusses about the engineering standards and design challenges faced during the time doing the project. This chapter also includes the discussion on the impact and ethical aspects of the project with complex engineering problem.

5.1 Compliance with the Standards

Compliance with engineering standards is crucial to ensure the reliability, scalability, and performance of the Bangla abstractive summarization models. The standards applied to this project relate to software tools, hardware, and communication protocols, and alternatives were evaluated to determine the best choices for development.

5.1.1 Software Standards

Python 3.12: The primary programming language used for this project due to its extensive support in the NLP and AI communities.

Transformers (Hugging Face)
The BanglaT5 and mT5 models were implemented and fine-tuned using Hugging Face transformers library. With support of numerous pre-trained models, the library smooths fine-tuning and deployment for natural language processing (NLP) tasks such as summarization.

Alternative Considered: OpenNMT
OpenNMT is a well-known sequence-to-sequence framework that provides a lot of flexibility and is very extensible. This library is that it is not very user-friendly, and it does not have the pretrained models that Hugging Face has, therefore its not a good candidate for our project.

PyTorch
The deep learning framework was PyTorch to train the models. Due to its dynamic computation graphs and greater flexibility, it is one of the most popular choice for research.

Alternative Considered: TensorFlow
TensorFlow: Another powerful deep learning framework, particularly when it comes to production. But for this project, I subjected: PyTorch is much easier to use and a better community support for NLP.

Datasets (Hugging Face):
Used Datasets library for easy manipulation and loading of BANS dataset. It includes a support for pre-processing and tokenization which makes it great at performing on large datasets quickly.

Alternative Considered: Pandas + Custom Dataloader
Pandas is an amazing library for data manipulation but on the other hand, Hugging Face has Datasets library which is designed to integrate seamlessly with Transformers and are easy to tokenize also.

Normalizer (CSE BUET):
The Bangla text normalizer from NLP lab of CSE BUET was used to clean and standardize Bangla text before tokenization. It addresses problems such as Unicode normalization and formatting discrepancies.

Alternative Considered: spaCy
While spaCy has a great text preprocessing with the exception of that spaCy does not have Bangla support, which makes the Bangla-specific normalizer the better choice than spaCy.

Rationale for Software Selection:
We chose Hugging Face Transformers and PyTorch for two reasons. They were industry standard tools the wider the net is cast, the more familiar people will be, and that becomes the case when we discuss tools that are user-friendly and well-supported Hugging Face is a dominant name in the NLP community these days.

We settled on PyTorch rather than TensorFlow for this project given our need for flexibility with an easy API especially since we were using a pre-trained model and working on a dynamic task (summarization).

Hugging Face Datasets was picked because it offers pre-built ways to work with plenty of NLP tasks and with big dataset and therefore, our data handling would take less work and time.

As a normalizer we had to choose, between some general normalizer tools (spaCy etc.) and Bangla-specific normalizer, we went for Bangla-specific one because those tools are not specifically designed for Bangla text, hence it will not totally work.

5.1.2 Hardware Standards

The hardware environment for the Bangla abstractive summarization project was selected to ensure a balance between performance, cost, and availability. Given the computationally intensive nature of deep learning tasks such as training BanglaT5, mT5, and GPT-4o, the hardware setup was optimized for both model development and large-scale model training.

Selected Hardware:

MacBook Air M1: The MacBook Air M1 was used for initial model development and experimentation. The Apple M1 chip provides significant performance improvements in both CPU and GPU tasks compared to previous Intel-based models, especially in energy efficiency. While this setup is not ideal for large-scale model training, it was suitable for smaller datasets, data preprocessing, and fine-tuning smaller models.

Specs:

Chip: Apple M1

RAM: 8GB

Storage: 256GB SSD

Use case: Initial development, prototyping, and smaller-scale experimentation.
Windows Machine (i5-12400, 16GB RAM):

For more substantial computational tasks, particularly model training and hyperparameter tuning, a Windows machine with an Intel i5-12400 processor and 16GB RAM was used. This machine provided a perfect balance of computational power and memory capacity to handle our project requirement of mid-sized models like mT5 and BanglaT5 before scaling up to cloud-based resources.

Specs:

Processor: Intel i5-12400

RAM: 16GB

Storage: 512GB SSD

Use case: Model training, hyperparameter tuning, and testing.

Kaggle T4 GPU: For deep learning model training and larger dataset, we used Kaggle GPU T4 listed here. Ideal for training and inference tasks, T4 balances price and performance with acceleration for deep learning applications at low power consumption. Mixed-precision training (FP16) It supports mixed-precision training (FP16) that is important for speed and memory consumption when dealing with large models

Specs:

GPU: NVIDIA T4 (16GB VRAM)

CUDA: Supports CUDA 11.2 and TensorFlow 2.x

Use case: Model training at scale, fine-tuning on large datasets, and multi-GPU support.

Alternative Hardware Considered:

AMD GPUs: For GPU based training, AMD GPUs were an option. AMD GPUs, while cheaper than NVIDIA, are not as deep learning friendly since TensorFlow/PyTorch (the most used deep learning libraries) support CUDA, NVIDIA's library for developers. Thus, NVIDIA GPUs were finally chosen, for both better performance and doability with deep learning frameworks.
Cloud Services (AWS, Google Cloud): AWS EC2 instances with NVIDIA V100 or A100 GPUs used to be the favorite for such massive training. On the other hand, Kaggles free infrastructure with T4 GPUs was a better deal and simpler to use for the scale of our project. This alternative cloud services were not taken due to costs and availability of free resources based on Kaggle.

Rationale for Selection:

- The combination of MacBook Air M1, Windows machine (i5-12400), and Kaggle T4 GPU was selected due to several factors:

- Performance vs Cost: The MacBook Air M1 and Windows machine offered a good balance for our task of model development and smaller-scale training without any hardware costs.

- Scalability: Kaggle's T4 GPU provided the necessary computational power for large-scale training without requiring additional investment in cloud services.

- Compatibility: The NVIDIA T4 GPU's compatibility with TensorFlow and PyTorch ensures efficient training and model fine-tuning, with added support for mixed-precision training.

- Resource Availability: The Tesla T4 GPU on Kaggle allowed for access to powerful hardware without the need for any investment in physical servers or cloud services.

5.1.3 Communication Standards

Effective communication between the various components of the Bangla Abstractive Summarization system is crucial to ensure smooth interaction and data flow. As the project utilizes Gradio for the web interface, and Hugging Face for hosting the model, it becomes essential to use a suitable communication protocol for secure, efficient, and interoperable data. The selected criteria for communication in the project are described as follows.

1. API Interaction

RESTful APIs were leveraged to enable a smooth line of communication between the frontend (Gradio interface) and the backend (Hugging Face model). Users can input Bangla articles and get abstractive summaries as output in the Gradio interface. Interface to Model: The interface to the model is through RESTapis, with the input data in the form of text send as json and model's output as the summary also returned as json.

Alternatives Considered: For reasons of comparing protocol performance and feature capabilities, use of a SOAP web service was considered as an alternative to RESTful API. SOAP was never chosen after all, because of its high complexity, strictness, and overhead.

While RESTful API design is simple, RESTful APIs are more lightweight and less rigid and makes them perfect fit for a project like this one.

Rationale for Selection: RESTful APIs were chosen due to their easiness, scalability, and popularity in web implementation methods. Moreover, given that JSON is used for data formatting, enabling an efficient and human-readable flow of communication between the frontend and the backend; it seemed like the right decision for this project.

2. Secure Communication

The exchanged information between the Gradio system and the Hugging Face model was safeguarded for confidentiality and integrity since HTTPS was used for communication. HTTPS encrypts the data transferred over the web making both user inputs (Bangla news article) and outputs (Summaries) secure.

Alternatives Considered: While HTTP could be used for communication, it lacks encryption and protection, making it not suitable for transmitting sensitive data given by the user. Given the importance of data security in this project, HTTPS was selected over HTTP.

Rationale for Selection: HTTPS was chosen because as it will secure the communication by encrypting the data sent between the client (frontend) and server (backend). This is important to secure data privacy and preserve correctness in Bangla summarization. This is significant as far as security and safe in Bangla summarization then particularly when the user data might be sensitive one.

3. Data Format: JSON

The data that is sent back and forth between the frontend and backend is in JSON format (JavaScript Object Notation). JSON is a light and easy-to-read format that is often used for data exchange in web apps. It is good for sending and receiving Bangla text and its summary because it is both space-efficient and easy to read.

Alternatives Considered: XML (Extensible Markup Language) was considered as an alternative to JSON. XML is also used for data exchange, but it is longer in format and less efficient than JSON, which makes it not considered for this project.

Rationale for Selection: JSON was chosen because it is small, structured, easy to parse, and works well many programming languages and popular frameworks. JSON is the best choice for this project because it is widely used in web development and is good at handling text-based data.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The released Bangla abstractive summarization models can greatly impact people's lives, especially the people of Bangla community by its development and application. By giving users short, relevant summaries of long articles and other text the system makes information more accessible, saving time and enhancing content consumption. Furthermore, it has a significant impact on digital skills attainment, allowing people to

access more content more effectively, and it promotes more engagement in the digital domain. The system also can provide an educational effect by enabling students and professionals to digest a large quantity of text, as well as being useful for the illiterate and the aged, enabling them to achieve various information via Internet and facilitating them in social life. Additionally, cross-lingual properties of models such as mT5 can achieve a wider variety of summarization tasks, promote the communication between different languages and reduce communication barriers. In sum, this project increases Bangla speakers' access to knowledge, and therefore to a more educated, digitally literate, and connected world.

5.2.2 Impact on Society & Environment

The abstractive summary system for Bangla could have a big effect on society by bringing together elite and regular Bangla speakers in the digital knowledge society. The project helps "democratize" information by making news and educational material easier for more people to find and understand. It does this by providing cheap, automatic production of short, accurate summaries. This is especially helpful in a society where people have trouble understanding each other and can't get to your information, like when Bangla is a low-resource language in NLP. It helps people who are on the outside of society or who are less educated by giving them tools that make it easier for them to learn. It also helps the environment by using cloud resources like Kaggle's T4 GPU and the scalable tech they bring, which reduces the need for hardware and has less of an impact on the environment. The truth is that the project is about getting more information, protecting the rights of less fortunate people, and making the environment more sustainable.

5.3.3 Ethical Aspects

As with any artificial intelligence (AI) project, particularly one involving natural language processing (NLP), it is important to consider the ethical implications of the models' development and deployment. The Bangla abstractive summarization project raises several ethical considerations, particularly around data privacy, bias in AI models, and responsibility in the use of automated content generation.

Data Privacy and Security: The main ethical problem with this research is the use of Bangla news articles and their summaries as data. Data used for model training, such as the BANS dataset, must adhere to very strict data privacy standards. That means that the dataset doesn't include anything personal or sensitive. Content where personal data is used in any form, consent and anonymization steps also have to be undertaken so that the privacy of individuals is protected.

Bias in the Model: The last ethical concern is the possible bias in the summaries that the models generate. Nevertheless, models such as BanglaT5 or mT5 are trained on social data that may contain biases of any kind, such as gender, political, cultural, or international. Furthermore, the summarization model may perpetuate these biases by producing summaries that either understate or distort the original article's content or fail to adequately cover the news. The development or training dataset must be carefully gathered and chosen to be representative and diverse in order to prevent these kinds of problems. Additionally, domain-specific experts must regularly verify that the models are producing fair results.

Accountability for Automated Summarization: Using automatic summarization tools to make news has effects on who is responsible. The models should make summaries that are true and don't lie to readers. We've seen that automated systems like this can make it easier to consume content, but they shouldn't be trusted completely. There needs to be a balance, and people need to look over and check the summaries, especially for topics that could be sensitive or important (politics, health, legal, etc.).
Transparency and Explainability: Like any AI model, we need to be able to explain and understand the decisions made by the model. This contributes to the trust in the system formation and is convenient for users to grasp the way for summaries to be generated. The users should be notified what the AI tool does, its flaws and that the model is not perfect all the time and might generate wrong or biased summery. Explainability also means trying to grasp how the model is deciding and to try to counteract potential sources of bad output.

5.2.3 Sustainability Plan

The Bangla Abstractive Summarization project is planning for environmental sustainability and the sustainability of the system. The project utilizes cloud-based computer resources, such as Kaggle's T4 GPU in order to greatly reduce the level of environmental cost involved by not needing high-end local hardware that requires a high level of energy. We are in addition making the training process resource efficient through mixed-precision training. The codebase is written following best-practices to be DRY, scalable, and easily updatable for future changes or improvements. The project also encourages open-source collaboration, with the releasing of models and datasets on places such as Hugging Face for everyone to learn and improve upon. Ethical data privacy and copyright laws are observed, using openly available datasets for the project. Further, the project aims to continually integrate in new Bangla data sets to ensure that the models remain up-to-date and performative as the language changes. For lessening its carbon impact, the project operates on efficient cloud infrastructure and not power-hungry local machines. This also ensures that the sustainability of the project is preserved in aspects of environmental sustainability as well as keeping the work up to date in the fast-changing field of Bangla NLP.

5.3 Project Management and Financial Analysis

This section outlines the estimated investment of time in developing and deploying the Bangla Abstractive Summarization system that uses deep learning models like BanglaT5, mT5, and GPT-40. The cost analysis includes software and hardware as well as R&D, operational cost, and deployment. Also, the minimum solution budget is proposed in this section and future revenue models for the sustainability of the system are investigated.

The initial budget for the project, which covers all necessary expenses for the development, training, deployment and maintain of the Bangla summarization models, is as follows:

Table 5.1: Financial Analysis

SN	Category	Amount (BDT)
01	Premium Tool (GPU)	3500-5000
02	Hosting	2000-2500
03	R&D	2500-3000
04	Operating Cost	1500-2000
05	Contingency	1000-1500
Total Estimated Cost		10500-14000

Alternative Budget (Minimal Solution): For a cost-effective budget solution, a minimal budget has been proposed, which uses free-tier resources and open-source tools:

Table 5.2: Financial Analysis (Minimal)

SN	Category	Amount (BDT)
01	Premium Tool (GPU)	Free
02	Hosting	2000-2500
03	R&D	1500-2000
04	Operating Cost	500-1000
05	Contingency	200-500
Total Estimated Cost		4200-6000

Revenue Model

For sustainable and scalable Bangla Abstractive Summarization service to address, several revenue models has been proposed:

Licensing Fees: Bangla Summarization tools can be licensed to institute es, news agencies, business organizations and educational institutions that want them for in - house or external use.

The Subscription Model: In the model, the system can generate the continuous revenue by having the user pay the subscription fee that gives access to the summarization services of the Bangla document for a month or a year.

Data Analytics Services: Providing data analytics services using the summaries produced by the models. Data relating to content trends, sentiment or user interactions can be sold to advertisers, research organizations and news media companies.

Consult and customize: Consultancy on implementation of the summarization tool into third party platform such as news website or educational organization, or customization of the solution for a specific user case.

Advertisements: The web interface could include advertisements, yielding additional profits through negotiated sponsorships (say with media, educational sites etc).

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

The Bangla Abstractive Summarization research tackles multiple complex engineering problems (EPs) in areas such as deep learning, natural language processing, and Large Language Models. The problems encountered were mapped to engineering categories as shown below:

Table 5.3: Mapping with Complex Engineering Problem.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✓	✓	✓	✓	×	×	✓

EP1 (Dept of Knowledge) is met as this program combines core knowledge with specialized knowledge. It integrates deep learning on model training with NLP techniques for text pre-processing and summarization as well as domain skills in Bangla language processing. Applying pretrained models such as BanglaT5, and mT5 needed understanding of transformer-based models and fine-tuning with regard to summarization.

EP2 (Conflicting Requirements Space) is satisfied as this work needs to resolve multiple conflicting requirements, e.g. high accuracy versus model efficiency. Large models are expensive to train but must be used in low-latency solutions deployed to real-time applications, thus this trade-off is of utmost importance when selecting and tuning a model.

EP3 (Depth of Analysis) is fulfilled, as this work focused on detailed model analysis based on the quantitative evaluation in BERTScore. The quality of summary was compared among BanglaT5 and mT5 and a human evaluation was performed on the summaries to check the coherence and the brevity.
EP4 (Familiarity of Issues) is satisfied. The project addressed unique challenges like Bangla text tokenization, morphological complexities, and issues arising from language-specific syntactical structures. These challenges were expected and dealt with using

customized preprocessing pipelines and domain-adapted models.

EP5 (Extent of Applicable Codes) is not satisfied this project does not strictly follow formal engineering standards (e.g., ISO, IEEE), it adheres to best practices in AI development and software engineering. Open-source libraries such as Hugging Face and PyTorch are extensively used for model implementation and deployment.

EP6 (Extent of Stakeholder Involvement) is not fully satisfied as no direct input from external stakeholders was included during development but we did evaluate our summary by professionals, the system is designed to serve end users, including researchers, students, and media organizations. Future iterations of the project may involve collaborations with external partners to refine the system's usability and functionality.

EP7 (Interdependence) is satisfied. The interdisciplinary nature of the project required coordination between NLP experts, software engineers, and linguists. Effective model development, preprocessing, and UI integration were all critical elements that relied on cross-disciplinary collaboration to ensure the project's success.

Mapping with Knowledge Profile

The knowledge required for addressing the engineering challenges in the Bangla Abstractive Summarization project is mapped to the Knowledge Profile. This shows the integration of engineering fundamentals, specialist knowledge, and research literature to solve complex problems.

Table 5.4: Mapping with knowledge Profile.

K1 Natural Science	K2 Mathematics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K7 Comprehension	K8 Research Literature
×	×	✓	✓	✓	✓	×	✓

K3 - Engineering Fundamentals: The project applies fundamental machine learning concepts including model fine tuning, training, and evaluation such as BERTScore.

K4 - Specialist Knowledge: The research required specialist knowledge in Bangla NLP, summarization models, and transformer-based architectures like BanglaT5 and mT5.

K5 - Engineering Design: The system architecture was designed to embed models in a web-interface using Gradio and Hugging Face, so that the summarization tool is available for the end users to use.

K6 - Engineering Practice: Best practices in software engineering were maintained, such as version control (GitHub), cloud-based serving (Hugging Face) and model evaluation best practices.

K8 - Research Literature: The project capitalizes on recent academic work in summarization models, Bangla NLP and evaluation metrics to see that state of the art models is employed.

5.4.2 Engineering Activities

This section maps the engineering activities involved in the project, demonstrating how the engineering solutions were implemented to address the complex challenges identified.

Mapping with Complex Engineering Activities

This section is designed to map the overall problem and EA's *(multiple).*

Table 5.5: Mapping with Complex Engineering Activities.

EA1 Range of re-sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓	×	✓	✓	✓

EA1 - Range of Resources
The project utilized cloud-based infrastructure (Kaggle T4 GPU), NLP libraries (e.g., Hugging Face), and large-scale datasets (e.g., BANS dataset) for model training and evaluation.

EA3 - Innovation
The integration of GPT-4o prompt engineering for zero-shot summarization and the fine-tuning of BanglaT5 and mT5 for Bangla-specific tasks represents an innovative approach to Bangla NLP.

EA4 - Consequences for Society and Environment
The system enhances information accessibility for Bangla speakers, contributing to digital inclusion. The project also minimizes environmental impact by leveraging cloud services with energy-efficient GPUs.

EA5 - Familiarity
The project used widely accepted NLP frameworks (e.g., PyTorch, Hugging Face) and Bangla language models, ensuring that the system is based on industry standards and can be easily understood or extended by other developers.

5.5 Summary

This chapter talked about the engineering standards, social issues, and design problems that came up while making our Bangla abstractive summarization system. To make the implementation more robust, repeatable, and easy to maintain, software standards (Python 3.12, PyTorch, Hugging Face Transformers) and software version control processes were put in place. Hardware When looking at the trade-offs between cost and performance, Kaggle's NVIDIA Tesla T4 GPUs for training and MacBook M1 and Intel i5-12400 machines for local development were two pieces of hardware that were part of their hardware standards. Communication protocols were maintained through secure web-based communication utilizing Gradio interfaces and Hugging Face Spaces for deployment, ensuring accessibility and user privacy.

In addition to technical standards, the chapter also talked about the system's social and moral effects. The model helps make Bangla information more available, works toward digital inclusion, and is cost-effective by using cloud servers-based GPU resources wisely. It is also a scalable approach. To lower the chances of bias and misuse, we thought about ethical issues like being open, fair, and using publicly available data responsibly. A sustainable idea was made to help the system's flexibility and the environment.

The chapter also talked about project management and financial analysis, giving an initial and an alternate minimal budget, and suggesting a revenue model for long-term success. Finally, the talk about difficult engineering problems and tasks put the project into well-known groups of problem-solving, knowledge profiles, and engineering activities. This map showed how the project fits with both domain knowledge, technical knowledge, and social concern in order to address a problem in the world.

In short, the chapter showed that the Bangla abstract summarization system is not only a technical advance, but it also follows professional, ethical, and social standards, as well as cost-effective and environmentally friendly ones. With examples from real life. These factors enhance authenticity, phasing, and utility within the industry, while situating the system within the broader context of engineering problem-solving.

Chapter 6

Conclusion

This chapter summarizes the overall contributions of the research, discusses its limitations, and provides directions for future work. It reflects on how the findings advance Bangla NLP and identifies areas for further improvement.

6.1 Summary

This research aimed to present a comparative performance study of large language models (LLMs) on Bangla abstractive text summarization using three states of the art systems, BanglaT5, mT5, and GPT (prompt-engineered). The motivation behind it is the scarcity of good summarization tools for Bangla, which is a low-resource language, and, in the context of the current research, to rectify the lack of language fairness in NLP.

The methodology involved fine-tuned BanglaT5 and mT5 on the Bengali Abstractive News Summarization (BANS) dataset, and evaluated the GPT with prompt engineering without fine-tuning. The models have been evaluated with automatic evaluation metric (BERTScore) and human annotated using three annotators to annotate the summaries for Relevance, Coherence and Conciseness.

The results showed a clear performance hierarchy:

BanglaT5 outperformed others on Bangla evaluation settings, achieving the highest BERTScore F1 (0.8174, lang="bn") and also performed convincingly better (89.4% in human rating Conciseness).

mT5 underperformed significantly, with an F1 of 0.5507, reflecting the limitations of multilingual pretraining in capturing Bangla-specific semantics.

GPT, while not fine-tuned, excelled in human-rated Relevance (83.5%) and Coherence (80.4%), but was penalized for verbosity with a lower Conciseness score (55.8%).

Collectively, these findings establish that BanglaT5 is the most effective model for Bangla abstractive summarization, while GPT offers complementary strengths in fluency and semantic coverage. The study not only benchmarks models but also highlights the trade-offs inherent in balancing relevance, fluency, and conciseness.

6.2 Limitation

While the study provides valuable insights, several limitations must be acknowledged:

- Dataset Constraints: The BANS dataset, though widely used, is limited in both size and domain (news articles). Its summaries are short, headline-like, which may not reflect the diversity of summarization needs across domains such as healthcare, law, or education. Consequently, the models' generalization capability remains untested in more varied contexts.

- Model Scope: The comparative analysis was restricted to BanglaT5, mT5, and GPT. Other emerging models such as Claude, LLaMA, Gemma, or Mistral were not evaluated, primarily due to resource constraints. Their inclusion could have provided a more comprehensive performance landscape.

- Evaluation Limitations: Automatic evaluation relied mainly on BERTScore, which, while powerful, does not fully capture factual consistency, fluency, or informativeness. Human evaluation, though valuable, was limited to three annotators, which restricts the robustness and generalizability of the judgments.

- Deployment Constraints: The system was deployed on a small scale using Gradio and Hugging Face Spaces. Large-scale user studies or real-world deployment scenarios were not conducted, limiting insights into usability, scalability, and end-user adoption.

- Bias and Variability: Both automatic and human evaluations are subject to bias. BERTScore favors models aligned with multilingual embeddings, while human annotators may differ in their subjective interpretation of conciseness or relevance.

6.3 Future Work

To address these limitations and extend the contributions of this research, several avenues for future exploration are recommended:

Dataset Expansion and Diversification: Constructing larger, domain-diverse Bangla summarization datasets that include long-form summaries and multi-document inputs would provide a stronger foundation for model training and evaluation.

Exploration of Emerging LLMs. Evaluating advanced open-source and proprietary LLMs such as LLaMA-3, Mistral, Claude, and GPT-4 on Bangla tasks would broaden the comparative scope and yield deeper insights into cross-model trade-offs.

Enhancement of BanglaT5: Improvements can be made through reinforcement learning with human feedback (RLHF), domain-adaptive pretraining, and controlled decoding strategies (e.g., length-penalized beam search) to balance conciseness with detail retention.

Optimizing GPT for Bangla: Prompt-engineering techniques such as few-shot prompting, chain-of-thought prompting, or hybrid pipelines (GPT → BanglaT5 compression) could enhance GPT's utility for Bangla summarization while controlling verbosity.

Hybrid Approaches: A promising avenue is a two-step pipeline where GPT generates a fluent draft and BanglaT5 compresses it into a concise summary. Such hybrid approach could combine the best of both models.

Scaling Human Evaluation: Larger scale human evaluation with diverse annotators coupled with qualitative error analysis (e.g., hallucinations, factual inconsistencies) will enhance the robustness and provide actionable insights for model improvement.

Practical Deployment: Future work should focus on developing mobile, website based. extension or on device button type summarization tools for journalists, students, and professionals, ensuring that research outcomes translate into real-world impact.

References

[1] A. M. Mitri, G. Saha, S. A. Lyngdoh, and A. K. Maji, "Abstractive Summarization of Khasi Texts using Pretrained Large Language Models," Procedia Computer Science, vol. 258, pp. 4117–4127, 2025, doi: https://doi.org/10.1016/j.procs.2025.04.662.

[2] A. Khan, F. Kamal, M. A. Chowdhury, T. Ahmed, T. Rahman, and S. Ahmed, "BanglaCHQ-Summ: An Abstractive Summarization Dataset for Medical Queries in Bangla Conversational Speech," pp. 85–93, Jan. 2023, doi: https://doi.org/10.18653/v1/2023.banglalp-1.10.

[3] T. Rehman, S. Das, Debarshi Kumar Sanyal, and S. Chattopadhyay, "An Analysis of Abstractive Text Summarization Using Pre-trained Models," arXiv (Cornell University), pp. 253–264, Jan. 2022, doi: https://doi.org/10.1007/978-981-19-1657-1_21.

[4] M. W. Bagus Dwi Satya, A. Luthfiarta, and M. N. Althoff, "Comparative Analysis of T5 Model Performance for Indonesian Abstractive Text Summarization," *SISTEMASI*, vol. 14, no. 3, p. 1092, May 2025, doi: https://doi.org/10.32520/stmsi.v14i3.4884.

[5] A. Mukherjee, "Developing Bengali Text Summarization with Transformer Base model - NORMA@NCI Library," Ncirl.ie, Jan. 2022, doi: https://norma.ncirl.ie/6232/1/adityamukherjee.pdf.

[6] G. E. Abdul, I. A. Ali, and C. Megha, "Fine-Tuned T5 for Abstractive Summarization," International Journal of Performability Engineering, vol. 17, no. 10, p. 900, 2021, doi: https://doi.org/10.23940/ijpe.21.10.p8.900906.

[7] T. T, M. Bohra, P. Dadure, and P. Pakray, "Comparative analysis of T5 model for abstractive text summarization on different datasets," SSRN Electronic Journal, 2022, doi: https://doi.org/10.2139/ssrn.4096413.

[8] Ilanchezhiyan, R. Darshan, M. Dhitshithaa, and B. Bharathi, "Text Summarization for Indian Languages: Finetuned Transformer Model Application." Accessed: Sep. 02, 2025. [Online]. Available: https://ceur-ws.org/Vol-3681/T8-5.pdf

[9] H. Jin, Y. Zhang, D. Meng, J. Wang, and J. Tan, "A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods," arXiv.org, Mar. 05, 2024. https://arxiv.org/abs/2403.02901

[10] J. Fang et al., "Multi-LLM Text Summarization," arXiv (Cornell University), Dec. 2024, doi: https://doi.org/10.48550/arxiv.2412.15487.

[11] M. Kabir, M. S. Islam, Laskar, M. T. Nayeem, B. M. Saiful, and E. Hoque, "BenLLMEval: A Comprehensive Evaluation into the Potentials and Pitfalls of Large Language Models on Bengali NLP," *arXiv (Cornell University)*, Jan. 2023, doi: https://doi.org/10.48550/arxiv.2309.13173.

[12] M. Abu and M. S. Islam, "Evaluating Large Language Models for Summarizing Bangla Texts," Openreview.net, 2024. https://openreview.net/forum?id=Z0zfZ4bn4x (accessed Sep. 02, 2025).

[13] P. K. Mondal, M. M. Rana, Bibakananda Roy Shuvo, Kawshik Ahmed Ornob, A. Sattar, and M. S. Rahman, "Low-Resource Language Summarization: A Study of Bangla Using T5 architecture," pp. 1–7, Feb. 2025, doi: https://doi.org/10.1109/ecce64574.2025.11013501.

[14] H. Mahmud, M. Hasan, F. R. Kabir, and Md. Zahiruddin Aqib, "A Systematic Literature Review of Similarity Analysis Techniques for Bangla Text," Zenodo, Jan. 2025, doi: https://doi.org/10.5281/zenodo.14730649.

[15] Sadik Yasin Eftee and Ajwad Abrar, "Evaluating the Effectiveness of Large Language Models in Multi-Document Summarization of Bangla News Articles," Jul. 19, 2025. https://www.researchgate.net/publication/393842416_Evaluating_the_Effectiveness_of_Large_Language_Models_in_MultiDocument_Summarization_of_Bangla_News_Articles.

[16] D. M. Lal, P. Rayson, K. P. Singh, and Uma Shanker Tiwary, "Abstractive Hindi Text Summarization: A Challenge in a Low-Resource Setting," ACL Anthology, pp. 603–612, Dec.2023, Accessed: Sep. 07, 2025. [Online]. Available: https://aclanthology.org/2023.icon-1.58/

[17] C. Sunitha, A. Jaya, and A. Ganesh, "A Study on Abstractive Summarization Techniques in Indian Languages," Procedia Computer Science, vol. 87, pp. 25–31, 2016, doi: https://doi.org/10.1016/j.procs.2016.05.121.

[18] A. Agarwal, S. Naik, and S. Sonawane,"Abstractive Text Summarization for Hindi Language using IndicBART." Available: https://ceur-ws.org/Vol-3395/T6-5.pdf.

[19] H. Nguyen, H. Chen, L. Pobbathi, and J. Ding,"A Comparative Study of Quality Evaluation Methods for Text Summarization," arXiv (Cornell University), Jun. 2024, doi: https://doi.org/10.48550/arxiv.2407.00747.

[20] A. Bhattacharjee, T. Hasan, W. U. Ahmad, and R. Shahriyar, "BanglaNLG: Benchmarks and Resources for Evaluating Low-Resource Natural Language Generation in Bangla," arXiv preprint arXiv:2205.11081, 2022.

[21] L. Xue et al., "mT5: A massively multilingual pre-trained text-to-text transformer," Oct. 2020, doi: https://doi.org/10.48550/arxiv.2010.11934.

[22] P. Bhattacharjee, A. Mallick, M. S. Islam, and None Marium-E-Jannat, "Bengali Abstractive News Summarization (BANS): A Neural Attention Approach," Advances in intelligent systems and computing, pp. 41–51, Dec. 2020, doi: https://doi.org/10.1007/978-981-33-4673-4_4.

[23] T. Zhang, V. Kishore, F. F. Wu, K. Q. Weinberger, and Yoav Artzi, "BERTScore: Evaluating Text Generation with BERT," Apr. 2019, doi: https://doi.org/10.48550/arxiv.1904.09675.

213-15-4292

ORIGINALITY REPORT

15%	9%	9%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	4%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
3	arxiv.org Internet Source	1%
4	Submitted to United International University Student Paper	1%
5	aclanthology.org Internet Source	1%
6	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	<1%
7	"Proceedings of International Conference on Trends in Computational and Cognitive Engineering", Springer Science and Business Media LLC, 2021 Publication	<1%
8	Bui Thanh Hung, M. Sekar, Ayhan Esi, R. Senthil Kumar. "Applications of Mathematics in Science and Technology - International Conference on Mathematical Appplications in Science and Technology", CRC Press, 2025 Publication	<1%
9	iarj.in Internet Source	<1%